



**Уральский  
федеральный  
университет**

имени первого Президента  
России Б.Н.Ельцина

**Институт  
фундаментального  
образования**

**Л. И. ДОЛИНЕР**

# ОСНОВЫ ПРОГРАММИРОВАНИЯ В СРЕДЕ PascalABC.NET

Учебное пособие



Министерство образования и науки Российской Федерации

Уральский федеральный университет  
имени первого Президента России Б. Н. Ельцина

Л. И. Долинер

# **ОСНОВЫ ПРОГРАММИРОВАНИЯ В СРЕДЕ PASCALABC.NET**

*Рекомендовано методическим советом УрФУ  
в качестве **учебного пособия** для студентов,  
обучающихся по направлениям подготовки  
230400 «Информационные системы и технологии»,  
090900 «Информационная безопасность»*

Екатеринбург  
Издательство Уральского университета  
2014

УДК 004.43PascalABC.NET(075.8)

ББК 32.973-018.2я73

Д64

Рецензенты:

Уральский технический институт связи и информатики (филиал) ФТОБУ ВПО «Сибирский государственный университет телекоммуникаций и информатики» (выписка из протокола № 1 от 11.09.2012 г. заседания кафедры общепрофессиональных дисциплин технических специальностей, завкафедрой канд. техн. наук, доц. Н. В. Будылдина);

д-р пед. наук, проф. Б. Е. Старченко (Уральский государственный педагогический университет)

Научный редактор – д-р пед. наук, проф. Г. А. Матвеева

**Долинер, Л. И.**

Д64 Основы программирования в среде PascalABC.NET : учебное пособие / Л. И. Долинер. – Екатеринбург : Изд-во Урал. ун-та, 2014. – 128 с.  
ISBN 978-5-7996-1260-3

Пособие является частью комплекса, предназначенного для обучения основам программирования на языке Object Pascal в среде PascalABC.NET. Комплекс включает, помимо данной книги, методические рекомендации для преподавателя, дополнительный задачник и набор учебных файлов, обеспечивающих полноценный и эффективный процесс обучения основам программирования. Предназначено для студентов, учителей в системе повышения квалификации, преподавателей информатики.

Рис. 7. Табл. 1.

УДК 004.43PascalABC.NET(075.8)

ББК 32.973-018.2я73

---

*Учебное издание*

**Долинер Леонид Исаевич**

## **Основы программирования в среде PascalABC.NET**

Подписано в печать 11.09.2014. Формат 70×100 1/16. Бумага писчая.

Плоская печать. Усл. печ. л. 10,32. Уч.-изд. л. 6,6. Тираж 50 экз. Заказ № 1506.

Издательство Уральского университета  
Редакционно-издательский отдел ИПЦ УрФУ  
620049, Екатеринбург, ул. С. Ковалевской, 5  
Тел.: 8 (343) 375-48-25, 375-46-85, 374-19-41  
E-mail: rio@urfu.ru

Отпечатано в Издательско-полиграфическом центре УрФУ  
620075, Екатеринбург, ул. Тургенева, 4  
Тел.: 8 (343) 350-56-64, 350-90-13  
Факс: 8 (343) 358-93-06  
E-mail: press-urfu@mail.ru

ISBN 978-5-7996-1260-3

© Уральский федеральный университет, 2014

## Оглавление

ПРЕДИСЛОВИЕ.....	5
ЧТО ТАКОЕ «ПРОГРАММИРОВАНИЕ» .....	8
ГЛАВА 1. Знакомство со средой PascalABC.NET .....	15
ГЛАВА 2. Простейшие программы на языке Паскаль .....	21
ГЛАВА 3. Организация диалога .....	27
ГЛАВА 4. Графика в языке паскаль (первое знакомство) .....	30
ГЛАВА 5. Операции с целыми и вещественными числами .....	35
ГЛАВА 6. Оператор ветвления .....	38
ГЛАВА 7. Оператор выбора.....	43
ГЛАВА 8. Цикл с параметром .....	46
ГЛАВА 9*. Рисование забора .....	49
ГЛАВА 10 <sup>0</sup> . Звездное небо и прочие странности .....	53
ГЛАВА 11. Цикл с параметром: типовые алгоритмы .....	56
ГЛАВА 12. Проектирование программ и процедуры .....	62
ГЛАВА 13. Циклы с условием .....	69
ГЛАВА 14. Построение графиков функций .....	73
ГЛАВА 15. Графические примитивы. Перья и кисти .....	77
ГЛАВА 16*. Работа с графическими объектами .....	81
ГЛАВА 17*. Событийное программирование.....	83
ГЛАВА 18. Основные операции со строковыми данными .....	86
ГЛАВА 19. Использование процедур .....	91
ГЛАВА 20. Работа с массивами: первые шаги.....	95
ГЛАВА 21*. Сортировка массивов .....	102

ГЛАВА 22*. Многомерные массивы .....	105
ГЛАВА 23. Подпрограммы-функции .....	109
ГЛАВА 24. Работа с текстовыми файлами: чтение и запись .....	113
ГЛАВА 25. Работа с данными из текстовых файлов .....	119
ГЛАВА 26. Основы работы с типизированными файлами .....	125

## Предисловие

Программирование сегодня – это бурно развивающаяся отрасль производства программных продуктов. В конце прошлого века общаться с компьютерами можно было лишь с помощью программирования, и поэтому программирование изучали фактически во всех учебных заведениях (в т. ч. в школах и вузах). Времена изменились, общаться с компьютерами можно уже с помощью готовых компьютерных программ, и нужда в массовом обучении программированию вроде бы отпала. Однако все оказалось не так просто. В современные прикладные пакеты включаются, как правило, дополнительные средства программирования, обеспечивающие расширение возможностей этих пакетов. Например, практически в любом пакете MS Office есть среда программирования VBA (Visual Basic for Applications), обеспечивающая расширение возможностей этого пакета; профессиональная работа с системой «1С Предприятие» требует постоянного программирования для настройки на потребности каждой фирмы. Речь уже ведется о новом подходе, в рамках которого программирование – это обязательная компонента подготовки специалистов, собирающихся профессионально работать в определенной сфере, в которой предполагается использование IT-технологий.

Программирование сегодня – это не только и не столько знание языка программирования. Прежде всего это знание *технологии* программирования, умение *проектировать* и *разрабатывать* программы и программные комплексы на основе этой технологии, умение *строить модели*, ставить задачи и иметь представление о коллективной разработке программных продуктов. Все это принято называть *культурой программирования*.

В данном пособии вы познакомитесь с языком программирования Паскаль (точнее, его разновидности – Object Pascal), с элементами *структурного программирования* (т. е. программирования без использования оператора безусловного перехода), с *технологией проектирования «сверху вниз»*, с *модульным программированием* (т. е. с разбиением программы на подпрограммы для удобства отладки и коллективной реализации) и *элементами объектно-ориентированного программирования*.

Материал пособия необходимо изучать последовательно. В книге предложено два вида материалов: теоретический, который размещен в электронном пособии, и практический, который следует выполнять на компьютере. Для выполнения практических работ необходима среда программирования PascalABC.NET (любой модификации), а также электронная поддержка практических работ, размещенная в папке Examples. Можно, конечно, использовать для обучения и другие среды (Turbo Pascal, Borland Pascal, Free Pascal, TNT Pascal и т. п.), однако материал подготовлен именно для названной среды. Среда бесплатная, и ее можно скачать с <http://PascalABC.NET/>. И хотя программирование на языке Паскаль изучают традиционно в среде Borland Pascal, мы сочли целесообразным воспользоваться средой PascalABC.NET (разумеется, с разрешения авторов среды С. С. Михалковича и М. Э. Абрамяна, доцентов Ростовского государственного университета). Наш выбор был обусловлен следующими причинами:

1) традиционные Турбо-среды (тот же Borland Pascal) являются DOS-средами, т. е. для работы в них нужно использовать стандарты операционной системы MS DOS, что во времена Windows является, по меньшей мере, анахронизмом, поскольку данные навыки вряд ли кому-то пригодятся в будущем;

2) возможности среды PascalABC.NET, в отличие от других сред, максимально приближены к Delphi, что позволяет более естественно перейти от программирования учебного к программированию визуальному и объектно-ориентированному;

3) графика PascalABC.NET практически совпадает с графикой Delphi;

4) среда PascalABC.NET содержит прекрасную и понятную справку на русском языке с множеством примеров;

5) кроме того, в среду включен компьютерный задачник с возможностью автоматической проверки решений.

В книге приняты следующие обозначения:



– символ используется для выделения определений или инструкций;



– так помечаются подсказки и важные замечания;



– при встрече с таким символом следует показать преподавателю (кон-

сультанту) результаты выполнения заданий. Если вы работаете самостоятельно, то пропустите помеченный этим символом текст;

- 0** – данным символом помечены простые разделы и задания. Если для вас они не составляют трудностей, то их можно пропускать. Если какие-либо сомнения есть, лучше выполнить;
- \*** – звездочкой помечены некоторые лабораторные работы, разделы и задания, изучение или выполнения которых представляет определенную трудность для начинающих;
- \*\*** – двумя звездочками помечены разделы и задания, являющиеся дополнительными для начинающих и обязательными для тех, кто планирует использовать данный программный продукт в профессиональной деятельности.



# ЧТО ТАКОЕ «ПРОГРАММИРОВАНИЕ»

## 1. Что такое «программа» и как ее исполняет компьютер

Компьютер представляет собой устройство для исполнения программ. «Мозгом» компьютера является процессор, который призван «понимать» и исполнять эти программы.



*Программа* – это последовательность предписаний (команд), записанных на языке, понятном исполнителю.

В нашем случае исполнителем является процессор. А что за язык, который этот процессор понимает? Как известно, компьютер хранит информацию, закодированную с помощью двоичных чисел. Программа для компьютера – это обычная информация, поэтому и она хранится в виде набора двоичных чисел. Но согласно определению это не простой набор чисел, а именно набор команд, понятных процессору. Сочетание команд, которые понимает процессор, и правил их написания принято называть машинным языком, или языком программирования низкого уровня. Например, предположим, что некий процессор понимает следующие операции:

00000001 – сложить;

00000111 – разделить;

00000100 – переслать из одной ячейки в другую.

Правило написания команды для современного процессора выглядит следующим образом:

*<код операции> <адрес первого операнда> <адрес второго операнда>.*

Здесь под операндом понимается некоторое значение, с которым производится операция; адрес операнда – это адрес блока памяти, в котором находится операнд.

В таком случае, например, фрагмент программы, определяющей среднее арифметическое четырех чисел, будет выглядеть следующим образом (см. таблицу):

Пример компьютерной программы в двоичных кодах


Код операции	Адрес 1	Адрес 2	Комментарии
00000001	00011000	00011001	Сложить числа, хранящиеся в блоках памяти по адресам 00011000 и 00011001, и поместить результат в блок памяти с адресом 00011001
00000001	00011001	00011011	Сложить полученную сумму из блока памяти 00011001 и третье число, размещенное в блоке памяти 00011011, и поместить результат в блок памяти с адресом 00011011
00000001	00011011	00011110	Сложить полученную сумму из блока памяти 00011011 и четвертое число, размещенное в блоке памяти 00011110, и поместить результат в блок памяти с адресом 00011110
00000111	00011110	00111110	Разделить полученную сумму на число, хранящееся по адресу 00111110, и поместить по этому же адресу результат
00000100	00111110	10111000	Переслать содержимое блока памяти с адресом 00111110 в блок памяти с адресом 10111000

Разумеется, у реальной программы есть специальный заголовок, по которому процессор определяет, что это именно программа, и на практике программы существенно сложнее простейшего примера, приведенного здесь.

Справедливости ради стоит отметить, что в некоторых редких случаях с помощью машинных кодов пишутся программы для специальных устройств (стиральные машины, управляемые ракеты и т. п.), в которые встроены относительно слабые процессоры и ограниченная оперативная память, но от которых требуется практически мгновенное исполнение команд. Программы, написанные с помощью машинных кодов, занимают существенно меньшую память, чем созданные с помощью других средств, и, как следствие, быстрее исполняются.

## 2. Языки высокого уровня. Компиляторы и интерпретаторы

Технология написания программ на машинных языках весьма трудоемка и для создания больших программ фактически непригодна. Именно поэтому стали придумывать *языки высокого уровня*, т. е. языки, с помощью которых можно было бы легко и удобно разрабатывать большие программы. Кроме того, подобные языки должны быть доступны большому кругу людей. Это значит, что в конечном итоге язык программирования должен максимально приблизиться к естественному языку человека. Пока идеал не достигнут, но языки программирования уже стали близки к естественным языкам. Но как в этом случае процессор понимает программы, написанные на этих языках? Да никак. Дело в том, что языки высокого уровня созданы для того, чтобы люди могли естественным образом сформулировать перечень действий, выполняемых компьютером. Так как компьютеры языков высокого уровня не понимают, придумали специальные программы, называемые трансляторами.

 *Транслятор* (translator) – это программа, предназначенная для перевода (трансляции) описания алгоритма с одного формального языка на другой.

Алгоритм создания программы теперь предусматривает использование шага трансляции (рис. 1).

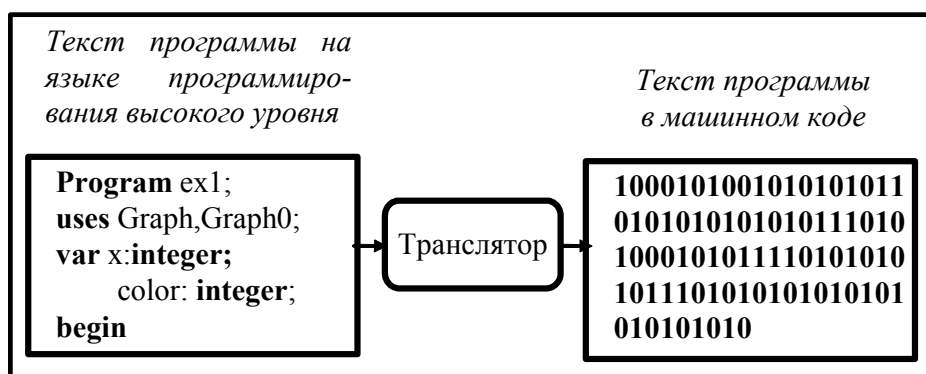



Рис. 1. Алгоритм трансляции программы


Этап превращения программы, написанной на языке высокого уровня, в машинный код реализуется в двух вариантах. В первом случае транслятор берет из файла программу на языке высокого уровня и переводит ее всю целиком в программу на машинном языке, создавая отдельный файл с расширением **EXE**. Про-


грамму, записанную в такой файл, принято называть *исполняемым модулем*, а транслятор, который выполняет такой перевод, называют *компилятором*.

Во втором случае транслятор берет из файла с программой на языке высокого уровня по одному предписанию (команде), транслирует ее в машинный код и передает полученную команду процессору для исполнения. Такой транслятор называют *интерпретатором*.

Существуют несколько сотен языков программирования высокого уровня. Каждый из них предназначен для решения определенного круга задач, например: BASIC и FORTRAN – для вычислений, PROLOG – для создания систем искусственного интеллекта, Pascal – для обучения программированию, COBOL – для обработки больших объемов данных, C++ – для профессионального программирования больших приложений и т. п.

 *Язык программирования* – это специально обусловленный набор символов, слов и мнемонических (особым образом организованных и заранее оговоренных) сокращений, используемых для записи набора команд (программы), воспринимаемых компьютером.

 *Синтаксис языка программирования* – это перечень правил записи программ из элементов этого языка.

 *Программирование* – это технология разработки программ с помощью языков программирования.

Одним из современных широко распространенных языков программирования является Паскаль. Он был создан на рубеже 1960–70-х годов швейцарским ученым Никлаусом Виртом и первоначально предназначался для обучения программированию. В настоящее время данный язык используется и для профессиональной разработки малых и средних компьютерных систем.

### **3. Технология создания программ**

Современное программирование фактически является промышленной технологией, с помощью которой создаются программы и компьютерные системы. Труд этот чаще всего коллективный, хотя не исключены некоторые успехи ин-

дивидуальных разработчиков. Например, MS Windows и MS Office – это результат работы нескольких сотен профессиональных программистов, аналитиков, проектировщиков, менеджеров и дизайнеров фирмы Microsoft (США). А, скажем, архиватор WinRAR реализовал индивидуальный разработчик Евгений Рошал (Россия).

Схематически технология создания программ проходит следующий ряд этапов:

а) постановка задачи, в рамках которой в общем виде описываются предполагаемые возможности программы;

б) проектирование программы, т. е. разработка структуры, интерфейса и детализированных возможностей будущей программы (вплоть до описания конкретных алгоритмов); в рамках проектирования реализуется разбиение программы на независимые подпрограммы (т. е. на функциональные фрагменты, представляющие собой некоторые макрокоманды типа «отредактировать файл», «отправить файл по электронной почте», «найти фрагмент» и т. п.);

в) программирование – на данном этапе задачи на написание подпрограмм раздаются программистам, которые реализуют описанные алгоритмы на языке программирования;

г) отладка и тестирование программ – это обязательный этап, позволяющий выяснить, делает ли программа (подпрограмма) то, на что она рассчитана, и насколько надежно она это делает; следует отметить, что важным этапом тестирования является проверка того, как данная подпрограмма работает совместно с другими подпрограммами, входящими в один проект.

В заключение следует отметить, что каждая программа имеет свой жизненный цикл, который тем больше, чем больше у программы возможностей по настройке и чем легче она модифицируется. Именно поэтому сейчас становится все более популярна технология, называемая OpenSource, предполагающая предоставление пользователям программы вместе с ее кодом на языке программирования высокого уровня.

#### 4. Системы и среды программирования

Изначально инструментарий программистов включал ряд средств разработки, в которые, помимо языка программирования, входили:

- специализированные или обычные текстовые редакторы, с помощью которых писались тексты программ;
- трансляторы, которые проверяли правильность соблюдения синтаксиса языка программирования и, если синтаксис не нарушен, преобразовывали текст программы в машинный код;
- специальные отладчики, которые позволяли, например, покомандно выполнять программу и смотреть получаемые результаты.

Сегодня подобными средствами никто не пользуется, поскольку разработаны более удобные интегрированные инструментальные среды, обеспечивающие выполнение полного комплекса взаимосвязанных работ по созданию программ. Фактически эти среды включают в себя перечисленные выше компоненты. Более того, среды постоянно совершенствуются и все более автоматизируют процесс создания программ.

Можно отметить три поколения подобных сред. К первому поколению относятся *Турбо-среды*, в которых фактически интегрированы специализированный текстовый редактор, транслятор и отладчик. Повышение производительности обеспечивалось тем, что в рамках одной среды можно было заниматься подготовкой, трансляцией и отладкой программ. К подобным средам можно отнести среды Borland Pascal 7.0, Turbo C и др.

Второе поколение – это *визуальные среды программирования*. Такие среды, помимо того что обладают всеми возможностями Турбо-сред, предоставляют разработчику огромное количество готовых фрагментов программ. Эти фрагменты сгруппированы в различные подменю в виде отдельных пиктограмм (иконки), и их включение в программу разработчика реализуется перетаскиванием этой пиктограммы в нужное окно с помощью мыши. Подобный подход позволил существенно увеличить скорость разработки программ, имеющих стандартные интерфейсы (кнопки, окна, обработчики событий и т. п.). К данному виду про-

граммного обеспечения можно отнести Delphi (язык программирования Object Pascal), Borland C++ Builder, JBuilder (язык Java) и др.

И, наконец, CASE-среды программирования, представляющие собой просто конструкторы программ, в рамках которых либо вообще не надо программировать (если не считать рисования каких-либо схем взаимодействия готовых компонентов), либо программировать лишь вид окон, обеспечивающих интерфейс готовой системы. Подобные среды обладают максимальной на сегодняшний день автоматизацией проектирования и реализации программ и позволяют наиболее быстро создавать различные специализированные информационные системы. Среди подобных средств можно назвать Vantage Team Builder (Westmount I-CASE), Designer/2000, ERwin+BPwin и CASE.Аналитик.

### **Контрольные вопросы**

1. Что такое программа?
2. В программе на машинном языке (см. таблицу на с. 9) выполняются операции с двумя операндами. А куда записывается результат?
3. Чем отличается машинный язык от языка высокого уровня?
4. Предположим, что в программе, описанной в таблице на с. 9, потребуется найти среднее арифметическое не четырех, а пяти чисел, причем пятое число будет храниться в блоке памяти по адресу 01010101. Как изменится текст программы?
5. В чем назначение транслятора? Чем отличается компилятор от интерпретатора?
6. Что такое язык программирования? Синтаксис языка программирования?
7. Какие этапы включает в себя технология создания программ?
8. В чем особенность Турбо-сред?
9. Чем визуальные среды программирования отличаются от Турбо-сред?
10. Каковы особенности работы в CASE-средах?

# Глава 1

## Знакомство со средой PascalABC.NET

Сначала подготовимся к работе.

### Задание 1

Попробуйте на диске найти папку Рабочая. Если таковой нет, создайте ее.

В папке Рабочая создайте папку Pascal, а в ней – папку с вашей фамилией.

Далее эту папку будем называть «Вашей папкой».

Из папки Учебная\Examples в свою папку скопируйте все имеющиеся там файлы.

### 1. Запуск среды

#### Задание 2

Запустите среду программирования PascalABC.NET (Пуск → Программы → PascaABC.NET → PascalABC.NET).

Если среда запустилась, перед вами появится окно (рис. 1).

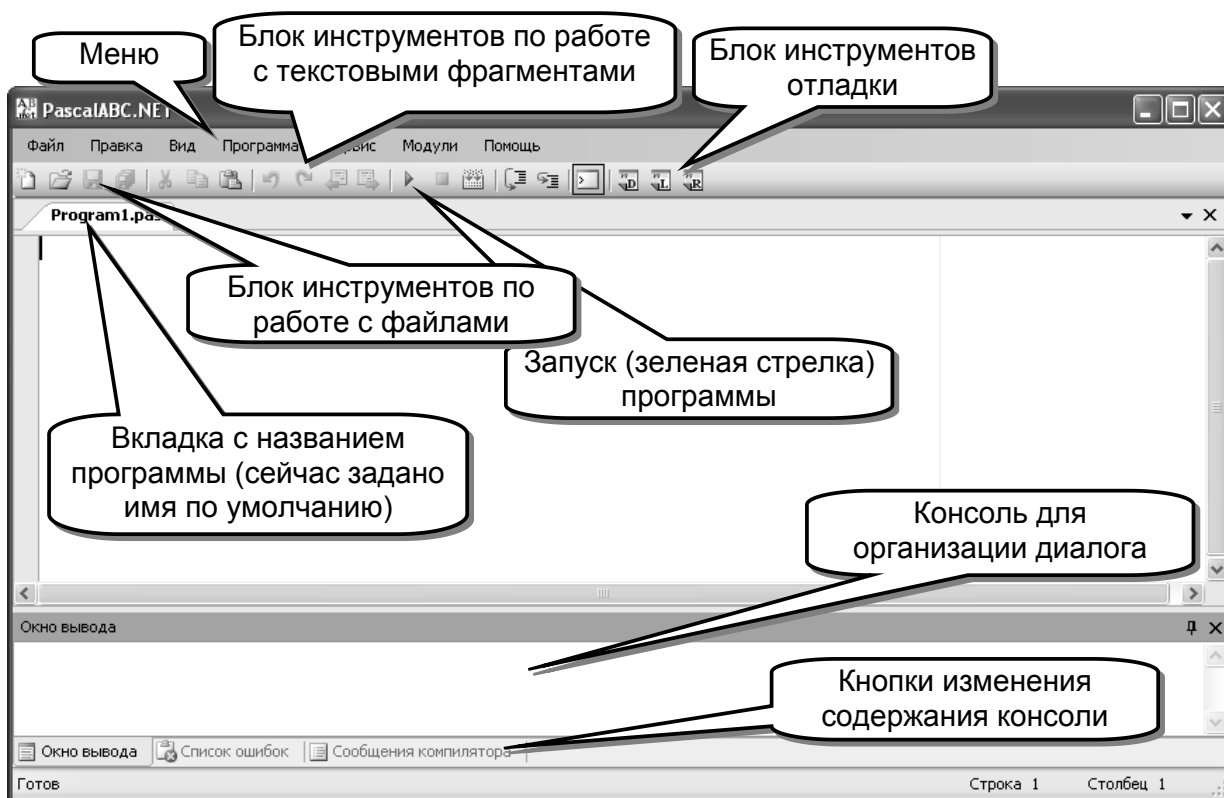


Рис. 1. Структура окна среды PascalABC.NET



### Задание 3

Изучите комментарии, приведенные на рис. 1.

Введем несколько определений, которые будем использовать в данном пособии.

*Консольные приложения* – это программы, в которых диалог организуется с помощью клавиатуры в специальном консольном окне (на рис. 1 это окно названо просто *консолью*). Окно может быть единым для операций ввода и вывода или делиться на две части (в нашем случае реализован последний вариант).

*Графические приложения* – это программы, выполняемые в отдельных окнах, в которых строятся различные изображения (в том числе и диалоговые окна), в которых диалог реализуется как с помощью мыши, так и с помощью клавиатуры.

## 2. Работа со страницами. Запуск программ

Лист, на котором будем размещать программу, будем называть *страницей*.

### 2.1. Как открыть программу и запустить ее на выполнение

#### Задание 4

Откройте программы **demo0.pas**, **demo1.pas**, **demo2.pas**, **colors.pas** из вашей папки. Программы (точнее, тексты программ) открываются обычным образом: Файл → Открыть или кнопка Открыть на стандартной панели инструментов.

Обратите внимание, что имена открытых файлов появляются на закладках в верхней части окна. Переход от программы к программе реализуется с помощью щелчка мыши на нужную закладку.

#### Задание 5



Запустите программу **demo0.pas**. Для запуска программы следует:

- перейти на страницу с текстом данной программы. Для этого достаточно щелкнуть по закладке с названием **demo0.pas** в верхней части рабочего окна;
- и щелкнуть по зеленой стрелке в панели инструментов;
- в случае, если программу следует прервать, нужно щелкнуть по значку в этой же панели, который находится справа от зеленой стрелки (рис. 2).

Убедитесь, что перед вами консольное приложение. Определите для этого приложения расположение окна вывода.

## 2.2. Как убрать окна ввода и вывода

### Задание 6

Скройте окно вывода. Для этого можно либо щелкнуть по кнопке Окно вывода () , расположенной в панели инструментов, либо просто нажать клавишу .

### Задание 7

Запустите программу **demo2.pas**. Ответьте на все вопросы программы.

Убедитесь, что перед вами консольное приложение. Определите для этого приложения расположение окон вывода и ввода. Если программа работает правильно (а она должна угадать любое задуманное вами число), то также закройте окно вывода.

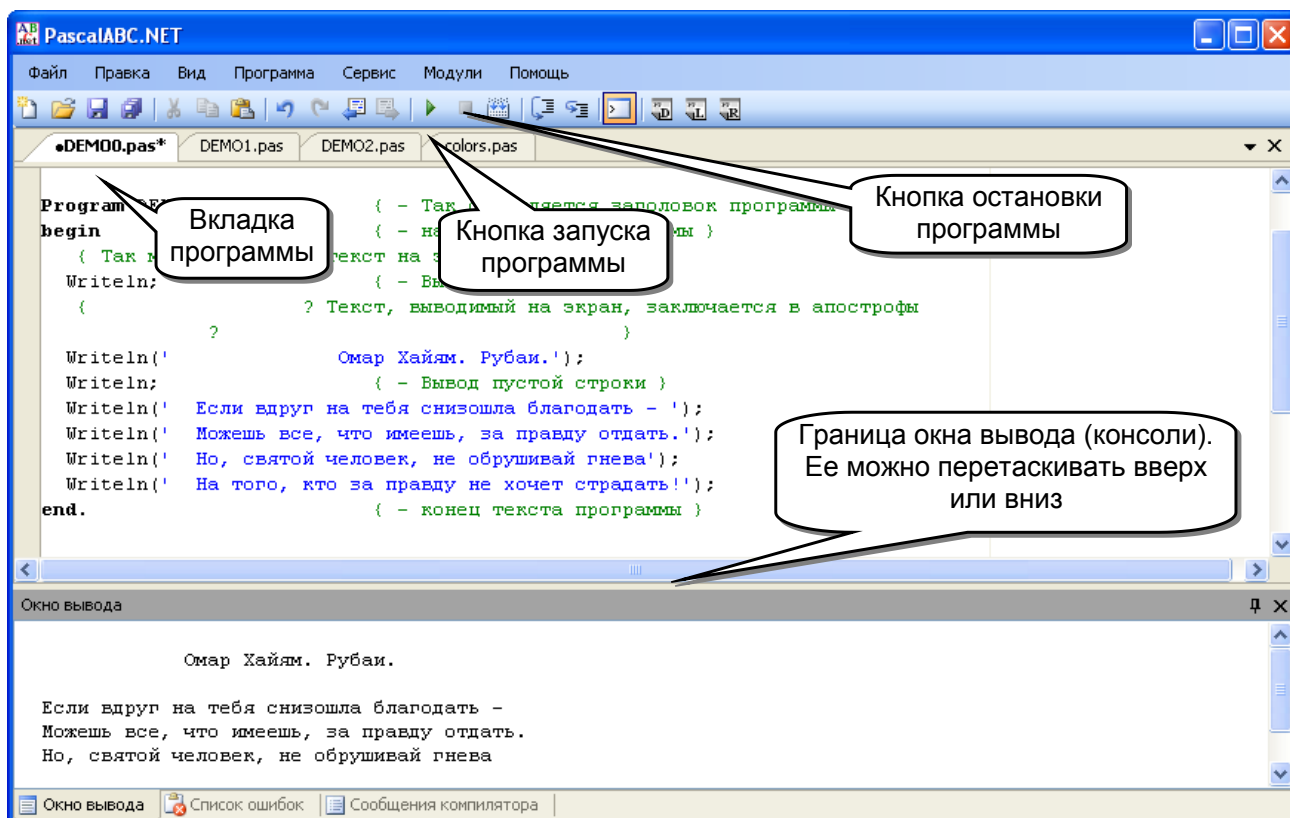


Рис. 2. Основные компоненты, используемые при запуске и остановке программы

☞ Чтобы было удобнее читать, расширьте окно вывода.

## Задание 8

Запустите программу **Colors.pas**. Попробуйте понять, как именно работает эта программа.

Убедитесь, что перед вами графическое приложение.

### 2.3. Как убрать имеющуюся страницу

Если изменений в текстах вы не производили, а текст больше не нужен для текущей работы, страницу можно убрать с рабочего стола. Для этого достаточно щелкнуть правой клавишей мыши по закладке и в контекстном меню выбрать либо пункт **Заккрыть** (т. е. закрыть текущее окно), либо **Заккрыть все**, кроме текущего, т. е. закрыть все окна, кроме текущего.

## Задание 9

Закройте все неактивные страницы. Выясните, можно ли закрыть оставшуюся страницу.

### 2.4. Как создать страницу

Новая страница – это новый документ. Для создания нового документа используется пункт меню **Файл → Новый**. В таком случае появится новая страница, а информация с этой страницы будет записываться в файл с именем **Program1.PAS**. В случае, если в новом листе началась разработка новой программы (т. е. просто начали набирать некий текст), то по попытке создать новую страницу будет создан файл с именем **Program2.PAS**, и т. д.

## Задание 10

Создайте новую страницу. После этого наберите текст следующей программы:

```
Program ex0;  
Begin  
  Write('Моя первая программа')  
end.
```

Запустите программу. Выясните, это приложение консольное или графическое.

☞ *Приложение становится графическим, если в нем подключается какая-либо библиотека.*

### 3. Среда PascalABC.NET как рабочий стол программиста

С точки зрения внешнего наблюдателя программирование – это процесс, схожий с писательским трудом. Программисту приходится много писать, опробовать созданное на «читателях» (компьютерах и людях, использующих эти компьютеры), выяснять, что тем «не нравится», исправлять написанное, снова проверять – и так до тех пор, пока не получится нечто, являющееся законченным продуктом.

Что требуется программисту для работы? Понятно, что стол, листы бумаги, ручка и компьютер с необходимыми программами. Все это и предлагает среда программирования PascalABC.NET.

Когда вы запускаете среду PascalABC.NET, получаете пустой «стол». На этот стол можно положить пустой лист «бумаги» (страницу) либо уже подготовленные «листы» (файлы с текстами). На этих «листах» можно писать любой текст. Для этого в среду PascalABC.NET включен текстовый редактор. Все тексты можно сохранить в файлах на диске.

Если текст представляет собой программу на языке программирования Паскаль, то можно дать команду на компиляцию (т. е. проверку правильности, а также преобразование в машинный код) и исполнение написанной программы. Ранее этот шаг мы называли «запустить программу». Дело в том, что в современных средах программирования шаг компиляции, проверки и запуска на выполнения объединены, и если ошибок в тексте программы нет, то программа действительно просто выполняется.

#### Задание 11

Сохраните первую вашу программу в вашей папке. Для этого выберите пункт меню Файл, там – режим Сохранить как..., найдите свою папку, задайте имя (например **FirstPrg.pas**) и щелкните по кнопке Сохранить.

## Задание 12

Снова откройте программы **demo0.pas**, **demo1.pas**, **demo2.pas** из вашей папки. Познакомьтесь с текстом каждой из программ.


Выясните, с помощью какой процедуры (команды) можно вывести текст на экран.


## Задание 13

Откройте электронное пособие (Пособие\index.html). Прочитайте там раздел «Что такое "программирование"». Ответьте на вопросы в конце раздела. Выполните тест и покажите результат преподавателю.

## Контрольные вопросы

1. Как запустить среду PascalABC.NET?
2. Как создать новую страницу? Как открыть текст ранее созданной программы?
3. Как закрыть текущую страницу с текстом программы? Можно ли закрыть все неактивные страницы?
4. Пусть у вас открыто несколько программ. Как сделать активной какую-либо из них?
5. Как исполнить программу, размещенную на какой-либо странице?
6. Какими словами ограничивается тело программы на языке Паскаль?
7. Какую команду (процедуру) следует использовать в программе для вывода текста?

 Если вы обучаетесь в компьютерном классе, пригласите преподавателя и продемонстрируйте ему вашу программу. Будьте готовы ответить на его вопросы. После этого сохраните все открытые программы и выйдите из среды PascalABC.NET.

 Если вы обучаетесь в условиях использования дистанционных образовательных технологий, просто вышлите файл с подготовленной вами программой преподавателю (если это предусмотрено технологической картой).

## Глава 2

# ПРОСТЕЙШИЕ ПРОГРАММЫ НА ЯЗЫКЕ ПАСКАЛЬ

## 1. Структура программы на языке Паскаль

Программа на языке Паскаль имеет следующую структуру:

```
program <имя>;  
<блок описания данных>  
Begin  
  <оператор 1>;  
  <оператор 2>;  
  ...  
  <оператор N>  
end.
```

- ☐ заголовок программы;
- ☐ начало исполняемой части программы;
- ☐ текст («тело») программы;
- ☐ конец программы.

Обратите внимание:

- 1) операторы (команды) отделяются друг от друга *точкой с запятой*;
- 2) *перед* словом **END** точка с запятой не ставится;
- 3) *после* слова **BEGIN** точка с запятой не ставится;
- 4) *в конце* программы после слова **END** обязательно ставится точка.


## 2. Простейшие операторы


### 2.1. Резервирование памяти для работы, или описание переменных


Самое главное действие, которое выполняет компьютер, – это запись информации в память. Суть действия проста: необходимо указать компьютеру, КУДА записывать и ЧТО записывать.

Для определения того, КУДА записывать, необходимо сначала дать указание компьютеру ЗАРЕЗЕРВИРОВАТЬ место в памяти. Выполняется это в *блоке описания данных* после слова VAR: **var num : integer**; оно представляет собой описание блока памяти с именем **num**, в котором можно хранить *целое* число (от -2 147 483 648 до 2 147 483 647). Слово «integer» (целое) после двоеточия указывает *тип* переменной (блока памяти) и означает, что требуется зарезервировать блок памяти для хранения именно целого числа. В таком случае говорится, что пе-

ременная `num` является переменной целого типа или переменной типа `integer` (читается как «интедже»). Имя блока памяти – вещь обязательная. Ведь в программе вам придется выполнять различные действия с содержимым этого блока. А чтобы указать компьютеру, с каким именно блоком вы хотите работать, требуется имя.


 *Имя блока памяти* – это последовательность букв и (или) цифр, начинающаяся с буквы. Имя может содержать и знак подчеркивания: `_`

 Блок памяти вместе с именем называют еще *переменной*, или *идентификатором*.

 **ВНИМАНИЕ!** Слова, зарезервированные в языке Паскаль, не могут использоваться в качестве идентификаторов!

Например, посмотрев на схему структуры программы, вы уже можете определить для себя несколько таких слов (список всех зарезервированных слов приведен в Помощи: раздел «Справочник по языку», там подраздел «Идентификаторы и служебные слова»):

**Program, begin, end, var, integer.**

 **ВНИМАНИЕ!** Блок описание данных *VAR* предлагает компьютеру выделить (зарезервировать) память для работы программы. При этом записи информации в блок **НЕ ПРОИСХОДИТ!**

Для начала вы можете воспользоваться следующими стандартными (т. е. «понимаемыми» компилятором) типами для описания переменных:

- integer** – для целых чисел в диапазоне от -2 147 483 648 до 2 147 483 647;
- byte** – то же от 0 до 255;
- word** – то же от 0 до 65 535;
- real** – для вещественных чисел;
- char** – для хранения одного символа типа Unicode (занимает 2 байта);
- string** – для строк текста произвольной длины.

## 2.2. Запись данных в память, или оператор присваивания

В предыдущем параграфе вам фактически предложено работать с данными трех типов: целыми, вещественными и строковыми. Следует запомнить:

1) целые числа в программах записываются так же, как принято в математике. Например: 345, -12222;


2) вещественные числа могут записываться двумя способами – первый – аналогичный математической записи: -123.234, 92929.3456 (обратите внимание, что здесь используется десятичная *точка*, а не запятая), – второй – в так называемом «плавающем» виде (правильное название: «десятичное число с плавающей точкой»). Например, число  $1,23 \cdot 10^{12}$  может быть записано следующим образом:


**1.23E+12**

Здесь «E+12» читается как «умножить на десять в двенадцатой степени»;

3) строки представляют собой произвольный набор символов, заключенный в апострофы:

‘Какой чудесный был пирог. Я от него ...  $2+2=5$ ’

 *Оператор присваивания* предназначен для указания компьютеру ЗАПИСИ ДАННЫХ В КОНКРЕТНЫЙ БЛОК ПАМЯТИ. Общий вид оператора:  
**<имя блока памяти>:=<выражение>**

 *Обратите внимание на знак присваивания – он состоит из двух значков: двоеточия и равно (:=), которые записываются друг за другом без пробелов.*

*Пример 1*

```
program example1;    {заголовок программы}
var num : integer;   {резервирование блока памяти для хранения целого
                        числа}
day : byte;           {резервирование блока памяти для хранения целого
                        числа от 0 до 255}
name : string;        {резервирование блока памяти для хранения строки}
begin
```



```

num:=355;           {запись в блок num числа 355}
day:=31;            {запись в блок day числа 31 }
name:='Скорпион'    {запись в блок name слова «Скорпион»}
end.

```

Как следует из определения оператора присваивания, в правой части оператора присваивания находится некоторое значение или *выражение*. В последнем случае в блок памяти (*переменную*) записывается *значение* выражения.

#### Пример 2

```

program
example2;
var    num    :
integer;
    day : byte;
Begin
    num:=355+145; {запись в блок num числа 500}
    day:=num -469; {при вычислении значения выражения вместо num будет
end.             подставлено значение из этого блока памяти (500) и вы-
                   числен результат (31)}

```

Допустимые арифметические операции языка Паскаль приведены в **Помощи**: раздел «Справочник по языку», там подраздел «Выражения и операции». Список стандартных функций находится в том же разделе, подраздел «Системные процедуры, функции, типы и константы».

### 2.3. Вывод данных на экран дисплея

Примеры программ, приведенных выше, иллюстрируют команду присваивания, но для вас их выполнение на компьютере бессмысленно, ведь программа записывает данные в память компьютера и заканчивает работу. Вы ничего не увидите. Чтобы увидеть, в языке есть процедура<sup>1</sup> **вывода WriteLn**.

#### Задание 14

Считайте с диска файл **lab2\_14.pas**. Разберитесь в тексте программы.

Выполните предложенные там задания.

---

<sup>1</sup> *Процедурой* называется программа, оформленная специальным образом и рассматриваемая как единая команда. Следует иметь в виду, что процедур вывода в PascalABC.Net достаточно много.

### Задание 15

Считайте с диска файл **lab2\_15.pas**. Разберитесь в тексте программы. Выполните предложенные там задания.

### Задание 16

Считайте с диска файл **lab2\_16.pas**. Выполните предложенные там задания.

### Задание 17

Считайте с диска файл **lab2\_17.pas**. Разберитесь в тексте программы и выполните предложенные там задания. Выпишите в тетрадь различие между процедурами **WRITE** и **WRITELN**.

## 2.4. Форматированный вывод информации

В окне вывода можно поместить строго определенное количество символов. Количество строк и количество символов в строке зависит от размера окна. Место для одного символа принято называть позицией.

При использовании процедур вывода **WriteLn (Write)** вам предоставляется возможность самим определять количество позиций, которое отводится для данного вида информации. Вывод данных с указанием числа позиций принято называть *форматированный вывод*.

### Задание 18

Считайте с диска файл **lab2\_18.pas**. Разберитесь в тексте программы. Уточните способы форматированного вывода целых чисел. Выведите целое число так, чтобы перед ним было соответственно 20, 30 и 40 пробелов.

### Задание 19

Считайте с диска файл **lab2\_19.pas**. Разберитесь в тексте программы. Уточните способы форматированного вывода строк. Измените программу так, чтобы выводилось еще две строки: первая – чтобы было всего 2 пробела перед текстом, вторая – в правой части окна.

### Задание 20

Считайте с диска файл **lab2\_20.pas**. Разберитесь в тексте программы. Отметьте в тетради способы форматированного вывода вещественных чисел. Измените программу так, чтобы выводилось еще два вещественных

числа: первое – 133.567, чтобы было 20 пробелов перед числом и 5 знаков после запятой; второе – 79954.23451 так, чтобы точка числа была под точкой предыдущего числа и три знака после запятой.

### Задание 21

Считайте с диска файл **lab2\_21.pas** и выполните предложенное там задание.


### Задание 22


Считайте с диска файл **lab2\_22.pas** и выполните предложенное там задание.

### Задание 23

Откройте **Помощь**, найдите там раздел «Справочник по языку», подраздел «Системные процедуры, функции, типы и константы», статью «Процедуры ввода-вывода». Прочтите статью.

Откройте файл **writeFormat.pas**, разберитесь в особенностях создания форматной строки.

 *Пригласите преподавателя и продемонстрируйте ему все выполненные вами задания. После этого сохраните все программы и выйдите из среды PascalABC.NET.*

 *Если вы обучаетесь в условиях использования дистанционных образовательных технологий, просто вышлите все файлы с подготовленными вами программами преподавателю (если это предусмотрено технологической картой).*

## Глава 3

# ОРГАНИЗАЦИЯ ДИАЛОГА

### Задание 24

В электронном пособии (Пособие\index.html) изучите раздел «Введение в язык программирования Паскаль». Ответьте на приведенные там вопросы. Выполните тест и продемонстрируйте оценку преподавателю.

### Команда ввода данных

Команда присваивания обеспечивает запись данных в память *непосредственно* в программе. Довольно часто требуется данные получить не во время написания программы, а во время ее работы. Для этого используется команда ввода READLN. Общий вид команды:

**READLN(<Список ввода>)**

Например:

```
var fantom : integer;
```

```
    story : string;
```

```
begin
```

```
    ...
```

```
    Write('Сколько у Вас увлечений:');
```

```
    ReadLn(fantom);
```

```
    ...
```

```
    WriteLn('Введите название любимого блюда:');
```

```
    ReadLn(story);
```

```
    ...
```

```
end.
```

### Задание 25

Считайте с диска файл **lab3\_25.pas**. Разберитесь в тексте программы. Запустите программу на выполнение. Постарайтесь понять, как работает процедура ввода. Выпишите в тетрадь алгоритм работы процедуры **READLN**.

### Задание 26

Считайте с диска файл **lab3\_26.pas**. Разберитесь в тексте программы. Выполните предложенные там задания.

### Задание 27

Считайте с диска файл **lab3\_27.pas**. Выполните предложенные там задания. Пригласите преподавателя и продемонстрируйте подготовленную программу.

☞ **ВНИМАНИЕ!** Не следует стирать в программе имеющиеся комментарии.

### Задание 28

Считайте с диска файл **lab3\_28.pas**. В этом файле содержится заголовок программы, оператор начала и конца программы. Вам следует из этой «болванки» составить программу, позволяющую определять количество денег, которое вы сможете получить через **N** месяцев, положив в банк **K** рублей под **P** процентов годовых.

### Задание 29

Считайте с диска файл **lab3\_29.pas**. В этом файле содержатся итоговые задания. Составьте предложенные программы в различных вкладках среды PascalABC.NET.

### Задачи для самостоятельной работы

1. Известны значения двух переменных, содержащие числа. Напишите программу, меняющую значения этих переменных между собой без использования третьей, буферной, переменной.
2. Вычислите объем и площадь поверхности параллелипипеда. Определите, какие данные должны быть введены, какие – выданы на экран.
3. Составьте программу определения корней квадратного уравнения, имеющего решения.
4. Составьте программу определения идеальной массы человека, если она считается по формуле:  $\text{идеальная масса (кг)} = \text{рост (см)} - 100$ .
5. Средняя наценка на товар составляет 18 %. Пусть известна розничная цена товара. Вычислите оптовую цену.
6. Даны координаты двух точек. Вычислите расстояние между ними.

7. Вы положили деньги в банк под 15 % годовых. Составьте программу определения общей суммы вклада через заданное число месяцев.
8. Длина некоторого отрезка составляет  $p$  метров. Напишите программу перевода ее в русскую неметрическую систему. Указание: 1 верста = 500 сажень, 1 сажень = 3 аршина, 1 аршин = 16 вершков, 1 вершок = 44,45 мм.
9. Составьте программу, которая вычисляет площадь равностороннего треугольника.
10. Напишите программу вычисления стоимости покупки, состоящей из нескольких карандашей, линеек и тетрадей. Их количество и цену задать вводом. Ответ вывести в виде:

*Сумма к оплате: ... р. ... к.*

11. Напишите программу, переводящую время, указанное в минутах, во время в часах и минутах.

### **Вопросы для контроля**

1. Для чего предназначена команда ввода?
  2. Чем отличаются и что общего у команд ввода и присваивания?
  3. В чем разница между командой присваивания и равенством в математике?
- ☞ *Пригласите преподавателя и продемонстрируйте ему все выполненные вами задания. После этого сохраните все программы и выйдите из среды PascalABC.NET. Будьте готовы ответить на вопросы.*
- ☞ *Если вы обучаетесь в условиях использования дистанционных образовательных технологий, просто вышлите все файлы с подготовленными вами программами преподавателю (если это предусмотрено технологической картой).*

## Глава 4

# ГРАФИКА В ЯЗЫКЕ ПАСКАЛЬ (ПЕРВОЕ ЗНАКОМСТВО)

Как было отмечено ранее, приложения (программы) в PascalABC.NET бывают консольные и графические. Если не подключать специальные библиотеки, работа будет выполняться исключительно в консольном режиме. Базовой библиотекой, обеспечивающей графический режим (но консольный при этом никуда не девается), является библиотека GraphABC. В этой библиотеке собраны три блока процедур и функция. Первый блок обеспечивает сам процесс рисования с помощью встроенных примитивов (т. е. простейших фигур). Второй блок позволяет загружать и выводить в графическое окно уже готовые рисунки. И, наконец, третий блок позволяет задать режимы работы процедур и функций двух предыдущих блоков.

### Подключение модуля GraphABC

Библиотека для любой среды программирования представляет собой набор готовых (т. е. ранее кем-то подготовленных) подпрограмм (процедур и функций), сгруппированных по определенным признакам. В частности, как уже отмечалось, в библиотеке GraphABC собраны компоненты, отвечающие за построение простейших изображений.

В программе библиотека подключается с помощью оператора **Uses**. Так как процедуры и функции рисования графики размещены в библиотеке GraphABC, требуется сначала подключить графическую библиотеку. Подключение данной библиотеки выполняется сразу после названия программы<sup>2</sup>:

```
Program <имя>;  
Uses GraphABC;
```

...

☞ *Поскольку название программы можно не писать, постольку собственно с подключения библиотеки и может начинаться любая программа.*

---

<sup>2</sup> На самом деле название программы писать не обязательно, поэтому обычно подключение библиотек – самая первая строчка программы.

### Задание 30

В Справке рассмотрите описание модуля GraphABC (Помощь → Справка → Стандартные модули → Модуль GraphABC). Изучите подраздел «GraphABC: графические примитивы». Выясните, с помощью каких процедур можно нарисовать окружность, прямоугольник и линию.

Просмотрите в этом же разделе статьи «GraphABC: подпрограммы для работы с пером» (с помощью этих подпрограмм можно задавать параметры линий) и «GraphABC: подпрограммы для работы с кистью» (эти подпрограммы позволяют задавать параметры закрашки внутренней части замкнутых примитивов – прямоугольников, эллипсов, секторов и т. п.).

Окно, в котором можно рисовать, может иметь разное количество точек как по вертикали, так и по горизонтали. Каждая точка имеет в окне свои координаты. Единственное, что все окна объединяет – это то, что в верхнем левом углу находится начало координат (рис. 3).



Рис. 3. Система координат в графическом окне

- ☞ Количество точек по вертикали и горизонтали зависит от размеров окна.
- ☞ Обратите внимание на то, что система координат в графическом окне отличается от привычной: ось  $Y$  направлена не вверх, а вниз (рис. 3)!



### Задание 31

Считайте с диска файл с именем **lab4\_31.pas**. Он представляет собой программу рисования линии на экране дисплея. Запустите эту программу – и вы увидите на экране синюю горизонтальную пунктирную линию. Модифицируйте программу так, чтобы:

- а) линия располагалась вертикально;
- б) линия располагалась по диагонали экрана;
- в) на экране рисовались две диагонали – одна красного цвета, а другая – желтого.

### Задание 32

Считайте с диска файл с именем **lab4\_32.pas**. Перед вами появилась программа рисования на экране дисплея квадрата и окружности, размеры которых вводятся с клавиатуры (запустите программу и убедитесь в этом). Модифицируйте программу так, чтобы с клавиатуры вводился только размер стороны квадрата. При этом окружность должна быть:

- а) вписана в квадрат;
- б) описана около квадрата.

☞ *Обратите внимание на то, что при организации диалога в графическом окне выводимая информация (процедура `WriteLn`) появляется в самом этом окне, а вводимая – в диалоговом окне. Поэтому при организации диалога в графическом окне следует пользоваться только процедурой `WriteLn`.*

### Задание 33

В файле **lab4\_33.pas** записана программа рисования заштрихованного квадрата, меняющего цвет после определенной паузы. Запустите ее и наблюдайте за ее работой. Модифицируйте программу так, чтобы цвет квадрата не менялся, а менялся тип штриховки.

Еще одна модификация: на экране, кроме квадрата, должен появляться еще и заштрихованный круг:

- а) того же цвета в другом месте экрана;
- б) другого цвета внутри квадрата.

### Задание 34

Нарисуйте на экране дисплея:

- а) российский флаг 300×200 пикселей (горизонтальные полосы: белая, синяя, красная – одинаковой ширины);
- б) французский флаг 300х200 пикселей; (вертикальные полосы: синяя, белая, красная – одинаковой ширины);
- в) японский флаг 300х200 пикселей (красный круг в центре белого прямоугольника).

### Задание 35

Модифицируйте составленные программы рисования:

- а) российского флага;
- б) французского флага;
- в) японского флага

– так, чтобы размеры и место на экране вводились с клавиатуры.

### Задание 36\*

Нарисуйте на экране дисплея простейшую модель Солнечной системы, изобразив планеты кружками различного цвета и диаметра, а их орбиты – окружностями. Усложните модель, добавив Луну, спутники Марса и т. п.

☞ Подробную информацию о процедурах и функциях модуля *GraphABC* можно прочитать в *Справке* (Помощь → Стандартные модули → Модуль *GraphABC*).

### Задачи для самостоятельной работы

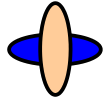
Составьте программу:

- 1) которая рисует закрашенное кольцо (●). Радиус внешней окружности следует задавать в диалоге;
- 2) выводит в центр окна вашу фамилию. Используйте для этого процедуру *TextOut*. При этом начертание должно быть полужирное, размер – 26 пт, цвет – зеленый;
- 3) рисует прямоугольник со скругленными углами. Необходимо, чтобы этот прямоугольник располагался в центре экрана, был закрашен любым цветом, толщина границы не менее 5 точек, цвет границы фиолетовый; цвет фона окна черный;

4) рисует три закрашенных сектора окружности, расположенных в центре графического окна следующим образом:



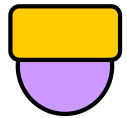
5) рисует два эллипса, расположенных в центре окна размером 250×250 точек следующим образом:



6) рисует три дуги толщиной 5 точек, яркого цвета, расположенные в окне 150×150 точек следующим образом:



7) рисует фигуру, состоящую из окружности и прямоугольника с закругленными углами. Толщина линий – 5 точек, яркого цвета, такой же штриховки, как на рисунке. Фигура расположена в центре окна 250×150 точек;



8) рисует треугольник в центре графического окна. При этом окно должно быть 200×200 точек, цвет фона – **clGold**, а треугольник нарисован с помощью процедур **MoveTo** и **LineTo**.

☞ *Пригласите преподавателя и продемонстрируйте ему все выполненные вами задания. После этого сохраните все программы и выйдите из среды PascalABC.NET. Будьте готовы ответить на вопросы.*

☞ *Если вы обучаетесь в условиях использования дистанционных образовательных технологий, просто вышлите все файлы с подготовленными вами программами преподавателю (если это предусмотрено технологической картой).*

## Глава 5

# ОПЕРАЦИИ С ЦЕЛЫМИ И ВЕЩЕСТВЕННЫМИ ЧИСЛАМИ

### Задание 37

Вспомните, какие операции можно выполнять с целыми числами.

Для этого:

- а) загрузите программу **lab5\_37.pas** и запустите ее на выполнение;
- б) запишите в тетрадь новую для вас информацию;
- в) составьте программу, которая позволит для любых заданных целых чисел найти разность квадратов и квадрат разности.

### Задание 38

Вспомните, какие функции преобразования вещественных чисел в целые есть в языке Паскаль. Для этого:

- а) загрузите программу **lab5\_38.pas**;
- б) запустите ее на выполнение;
- в) запишите в тетрадь новую для вас информацию;
- г) составьте программу, которая позволит определить доходы от вклада в два банка. Программа должна обеспечивать следующий диалог:

Здравствуйте. Задайте размер Вашего вклада:	120000
На сколько месяцев Вы делаете вклад:	10
Сколько процентов годовых Вам обещано в банке 1:	15
Сколько процентов годовых Вам обещано в банке 2:	17
Через 10 месяцев на Вашем счету в банке 1 будет	135000
Через 10 месяцев на Вашем счету в банке 2 будет	137000

### Задание 39

Вспомните, какие операции можно выполнять с вещественными числами.

Для этого:

- а) загрузите программу **lab5\_39.pas** и запустите ее на выполнение;
- в) запишите в тетрадь новую для вас информацию;

- г) составьте программу, которая по заданным сторонам треугольника определяет его площадь (напомним, что это можно сделать с помощью формулы Герона

$$S = \sqrt{p(p-a)(p-b)(p-c)},$$

где  $p$  – полупериметр;  $a, b, c$  – стороны треугольника);

- д) результат выведите, используя форматированный вывод.

## Задачи для самостоятельной работы

1. Составьте программу:

- а) которая определяет целое количество недель, которое осталось до Нового года. Для простоты считать количество дней в месяце равным 30 дням;
- б) вводит двузначное число и определяет сумму его цифр;
- в) вводит трехзначное число и выводит цифры этого числа в обратном порядке;
- г) вводит трехзначное число и определяет сумму его цифр.

2. Даны координаты вершин треугольника. Вычислите его площадь и периметр.
3. Напишите программу, переводящую время, указанное в минутах, во время в часах и минутах.
4. Ипотека обычно берется на большое количество лет (10, 15 и более). Составьте программу вычисления суммы, которая будет выплачена банку (срок выберите сами, скажем, 15 лет), если в год предполагается платить, кроме части долга, еще и 15 % годовых.
5. Длина отрезка задана в дюймах. Один дюйм равен 2,54 см. Перевести значение длины в метрическую систему, т. е. выразить ее в метрах, сантиметрах и миллиметрах. Например, 21 дюйм = 0 м 53 см 3,4 мм.
6. В языке Паскаль отсутствует стандартная операция «возведение в степень». Это сделано потому, что при определенной математической подготовке можно получить требуемый результат, используя имеющиеся возможности языка. Пусть нам необходимо возвести число  $a$  в степень  $x$ :  $y = a^x$ . Возьмем натуральный логорифм от обеих частей равенства:

$\ln y = \ln a^x \Rightarrow \ln y = x \ln a$ . Теперь воспользуемся этим равенством:  $e^{\ln y} = e^{x \ln a} \Rightarrow y = e^{x \ln a}$ . Запись этой формулы на языке Паскаль будет выглядеть следующим образом:  $y:=\exp(x*\ln(a))$ .

Напишите программу, которая возводит любое число в любую степень. Проверьте правильность вычислений по этой формуле для положительных и отрицательных чисел. Какие ограничения накладываются на вводимые данные?

7. Составьте программу, которая выводит на экран день недели по дате. Воспользуйтесь следующим алгоритмом вычисления дня недели:

день недели = остаток от деления  $X$  на 7,

где  **$X = \text{abs}(\text{trunc}(2.6*m-0.2)+d+y/4+y+c/4-2*c)$** ;


$m$  – номер месяца (см. ниже);


$d$  – число (день месяца);

$y$  – номер года в столетии;

$c$  – номер столетия (см. ниже).

Формула верна для григорианского календаря нового стиля (от 1582 до 4903 года). Год и месяц следует предварительно преобразовать так, как если бы начало года приходилось на 1 марта. Иными словами, март в этой формуле имеет порядковый номер 1, апрель 2, ..., январь 11 и февраль 12, причем январь и февраль следует отнести к предыдущему году. Например, для 1 февраля 1991 года номер месяца должен быть равен 12, а год 1990, в то время как для 31 декабря 1991 года номер месяца – 10, а год – 1991. Результат вычисления дается в виде целого числа в диапазоне от 0 до 6, причем 0 соответствует воскресенью.

 *Пригласите преподавателя и продемонстрируйте ему все выполненные вами задания. После этого сохраните все программы и выйдите из среды PascalABC.NET. Будьте готовы ответить на его вопросы.*

 *Если вы обучаетесь в условиях использования дистанционных образовательных технологий, просто вышлите все файлы с подготовленными вами программами преподавателю (если это предусмотрено технологической картой).*

## Глава 6

# ОПЕРАТОР ВЕТВЛЕНИЯ

### Задание 40

В электронном пособии (Examples\Пособие\index.html) изучите раздел «Условный оператор». Ответьте на приведенные там вопросы. Выполните тест по теме и продемонстрируйте оценку преподавателю.

### 1. Ветвление и его организация в языке Паскаль

При разработке алгоритмов очень часто возникают ситуации, когда выполняемые действия зависят от некоторого условия, т. е. сначала требуется проверить условие (или несколько условий) и в зависимости от того, выполняется оно или нет, выполнить различные действия. Например, при начислении стипендии проверяется, как студент сдал сессию. Если у него остались «хвосты», то ему будет назначена нулевая стипендия, если он сдал все на «отлично», то – повышенная, в остальных случаях – обычная.

Для организации таких действий в языке Паскаль имеется оператор ветвления (условный оператор, оператор IF).

### 2. Условный оператор IF

#### Задание 41

Запустите среду PascalABC.NET. Войдите в Помощь, в справочнике найдите раздел «Справочник по языку», подраздел «Операторы», в нем статью «Условный оператор». Просмотрите предложенный материал.

#### Задание 42

Запустите среду PascalABC.NET. Загрузите файл **lab6\_41.pas**. Прочитайте и проанализируйте текст программы. Запустите программу на выполнение и проверьте ее работу для различных данных. Для каких данных программа выдает некорректный результат? Модифицируйте программу таким образом, чтобы она в случае ввода одинаковых чисел выдавала сообщение: «Числа равны!»

Покажите результат преподавателю.

### Задание 43

Психологи рекомендуют учитывать оптимальное соотношение возрастов при вступлении в брак. Оно таково: возраст невесты равен половине возраста жениха плюс 7. Составьте программу для определения, подходят ли жених и невеста друг другу по возрасту.

### Задание 44

1. Составьте программу, которая позволяет найти максимальное из трех заданных чисел.
2. Протестируйте эту программу (проверьте корректность ее работы в разных случаях). Сколько наборов входных данных нужно проверить? Запишите их в тетрадь.
3. Загрузите файл **lab6\_44.pas**. В данной программе реализован оптимальный, на наш взгляд, алгоритм нахождения максимального числа. Сравните этот алгоритм со своим. Попробуйте оценить, какой из алгоритмов является оптимальным. (Оптимальным мы считаем тот алгоритм, который позволяет либо сократить время выполнения программы за счет уменьшения количества выполняемых команд, либо уменьшить объем памяти за счет уменьшения количества переменных.)

### Задание 45

1. Загрузите файл **lab6\_45.pas**.
2. Прочитайте и проанализируйте текст программы. Обратите внимание на составное условие.
3. Модифицируйте эту программу так, чтобы в том случае, когда введены значения сторон (такие, что из них можно построить треугольник, помимо выдачи текста «можно»), еще подсчитывался периметр треугольника.

### Задание 46

Составьте программу, которая умеет решать квадратные уравнения (по заданным коэффициентам уравнения  $A$ ,  $B$ ,  $C$  находит корни уравнения). Рассматриваются различные случаи: нет корней, есть один корень, два различных корня.



### Задание 47

По заданной стороне квадрата и радиусу круга выяснить, поместится ли:

- а) круг в квадрат;
- б) квадрат в круг.

Результаты вывести на экран.

### Задание 48

Составьте программу, которая умеет решать линейные уравнения (уравнения вида  $Ax + B = 0$ ). Рассматриваются различные случаи: нет решений, решением является любое  $x$ , единственное решение (вычисляет его).

### Задачи для самостоятельной работы

1. Программист должен был написать программу для решения неравенства  $ax + b > 0$  относительно  $x$  для любых чисел  $a$  и  $b$ , введенных с клавиатуры. Все числа считаются действительными.

```
var a, b, x: real;  
begin  
  readln (a,b,x);  
  If a=0 then  
    if b> 0 then write('любое число')  
      else write ('нет решений');  
    else write('x>','-b/a')  
  end.
```

Программист допустил ошибку в программе.

Последовательно выполните три задания.

- 1) Укажите числа  $a$ ,  $b$ ,  $x$ , при которых программа неверно решает поставленную задачу.
- 2) Укажите, какая часть программы является лишней.
- 3) Укажите, как нужно доработать программу, чтобы не было случаев неправильной работы (это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

2. Требовалось написать программу, которая определяет, существует ли треугольник со сторонами  $a$ ,  $b$  и  $c$ . Программист торопился и написал программу неправильно.

**Var a,b,c:real;**

**Begin**

**Write('a='); Readln(a);**

**Write('b='); Readln(b);**

**Write('c='); Readln(c);**

**If (a+b)>c then**

**If (a+c)>b then Writeln('треугольник существует')**

**Else Writeln('треугольник не существует');**

**End.**

Последовательно выполните следующее.

- 1) Приведите пример таких чисел  $a$ ,  $b$ ,  $c$ , при которых программа неверно решает поставленную задачу.
  - 2) Укажите, как надо доработать программу, чтобы не было случаев неправильной работы (можно любым способом)
  - 3) Напишите программу, в которой вложенные условные операторы замените условным оператором с логическими операциями OR, AND.
3. Дан массив натуральных чисел. Составьте программу, которая позволяет подсчитать и вывести среднее арифметическое элементов массива, имеющих нечетное значение.
4. Напишите программу, которая определяет, попадает ли точка с координатами  $(x, y)$  в круг радиусом  $R$  и центром в начале координат.
5. Напишите программу, которая определяет, в какой координатной четверти (квадрант) находится точка с координатами  $(x, y)$ . Выведите на экран номер квадранта. Учтите случай, когда точка лежит на одной из координатных осей или в начале координат.
6. Напишите программу, которая определяет, принадлежит ли точка с координатами  $(x, y)$  прямой  $y = 1 - x$ . Если это имеет место, то программа должна

вывести на печать координаты точки, в противном случае – сообщение: «Точка не принадлежит прямой».

7. Известны координаты трех точек. Напишите программу, которая определяет, находятся ли точки на одной прямой.
8. Напишите программу, которая определяет по введенным значениям длин сторон квадрата и радиуса круга, верно ли утверждение: «Круг вписан в квадрат».
9. С клавиатуры вводится трехзначное число. Напишите программу, которая проверяет, равна ли сумма его цифр значению 9.
10. Дано трехзначное число. Определить, равен ли квадрат этого числа кубам его цифр.
11. Даны два числа  $a$  и  $b$ . Составьте программу, которая определяет, является ли первое число делителем второго. А наоборот?
12. Дано трехзначное число. Составьте программу, которая определяет, есть ли среди его цифр одинаковые.
13. Год является високосным, если его номер кратен 4, однако из кратных 100 високосными являются лишь кратные 400 (например, 1700, 1800 и 1900 – невисокосные, а 2000 – високосный). Составьте программу, которая вводит число и определяет, является ли год с таким номером високосным.
14. Составьте программу, которые вводит три числа и определяет, является ли треугольник со сторонами данной длины прямоугольным.

☞ *Пригласите преподавателя и продемонстрируйте ему все выполненные вами задания. После этого сохраните все программы и выйдите из среды PascalABC.NET. Будьте готовы ответить на его вопросы.*

☞ *Если вы обучаетесь в условиях использования дистанционных образовательных технологий, просто вышлите все файлы с подготовленными вами программами преподавателю (если это предусмотрено технологической картой).*

## Глава 7

# ОПЕРАТОР ВЫБОРА

### 1. Цель использования оператора выбора

В ряде задач возникают ситуации, при которых нужно использовать оператор ветвления, поскольку в зависимости от некоего условия нужно выполнять то или иное действие, но при этом число условий достаточно большое. Например, если необходимо определить название для недели по его номеру, то решение задачи с помощью оператора ветвления будет выглядеть не очень красиво:

```
...
Write('Номер дня недели = '); Readln(n);
If n=1 then writeln('понедельник') else
    if n=2 then writeln('вторник') else
        if n=3 then writeln('среда') else
            if n=4 then writeln('четверг') else
                if n=5 then writeln('пятница') else
                    if n=6 then writeln('суббота') else
                        if n=7 then writeln('воскресенье')
                            else writeln('неверный номер');
...
```

Можно, конечно, решать подобные задачи рассмотренным способом. Однако для описания подобных действий в языке Паскаль имеется оператор выбора (оператор CASE).

#### Задание 49

В электронном пособии (Пособие\index.html) прочитайте раздел «Оператор CASE». Ответьте на приведенные там вопросы. Выполните тест и продемонстрируйте оценку преподавателю.

### 2. Оператор выбора CASE

#### Задание 50

Запустите среду PascalABC.NET. Войдите в **Помощь**, в справочнике найдите раздел «Справочник по языку», подраздел «Операторы», в нем статью «Оператор выбора». Повторите предложенный там материал.

#### Задание 51

1. Загрузите файл **lab7\_48.pas**. Прочитайте текст программы и проанализируйте его. Запустите программу на выполнение и посмотрите, как она работает.

2. Модифицируйте данную программу таким образом, чтобы она при вводе числа 0 выводила сообщение черным цветом (clBlack), а при вводе числа 8 – розовым (clPink).

### Задание 52

Составьте программу, которая позволяет для любых двух заданных чисел  $A$  и  $B$  и операции («+» – сложение, «-» – вычитание, «\*» – умножение, «/» – деление) получать на экране результат.

#### Подсказка

В данной задаче для хранения знака операции лучше всего использовать переменную типа **char**.

### Задание 53

Составьте программу, которая позволяет определить, является введенная буква гласной или согласной.

### Задание 54

Составьте программу, которая позволяет по возрасту определить, к какой группе относится данный человек (моложе 3 лет – совсем маленький, от 4 до 7 – детсадовский возраст, следующие 7 лет – подросток, от 15 до 20 – самый цвет и т. д. по вашему усмотрению).

### Задание 55\*

Известно, что астрологи делят год на 12 периодов и ставят каждому из них в соответствие знак зодиака:

20.1 – 18.2	⇒ Водолей	23.7 – 22.8	⇒ Лев
19.2 – 20.3	⇒ Рыбы	23.8 – 22.9	⇒ Дева
21.3 – 19.4	⇒ Овен	23.9 – 22.10	⇒ Весы
20.4 – 20.5	⇒ Телец	23.10 – 22.11	⇒ Скорпион
21.5 – 21.6	⇒ Близнецы	23.11 – 21.12	⇒ Стрелец
22.6 – 22.7	⇒ Рак	22.12 – 19.1	⇒ Козерог


Составьте программу определения знака зодиака для любого человека.


#### Подсказка

Идеальное решение для данной задачи – преобразовать дату в одно число: месяц умножить на 100 и прибавить день. Тогда, например, для Водолеев можно указать точный диапазон: 120..218.

## Задачи для самостоятельной работы

1. Составить программу, которая вводит номер месяца и выводит название времени года, к которому относится этот месяц.
2. Составить программу, которая вводит номер года и номер месяца, а выводит количество дней в этом месяце. Предусмотреть ситуации, когда год високосный, когда – нет (пояснения см. в задаче 13 для самостоятельной работы на с. 42).
3. Игральным картам условно присвоены следующие порядковые номера в зависимости от их достоинства: «валету» – 11, «даме» – 12, «королю» – 13, «тузу» – 14. Порядковые номера остальных карт соответствуют их названию («шестерка», «девятка» и т. п.). Составьте программу, с помощью которой можно будет определить карту по ее номеру.
4. Составить программу, которая вводит числовое значение оценки (от 1 до 5) и выводит ее название (1 – «плохо», 2 – «неудовлетворительно», 3 – «удовлетворительно», 4 – «хорошо», 5 – «отлично»).
5. Дано целое число в диапазоне от 10 до 40, определяющее количество заданий теста. Вывести текстовое описание заданного количества заданий, обеспечив согласование числа со словами «учебное задание», например: 18 – «восемнадцать учебных заданий», 21 – «двадцать одно учебное задание».
6. Дано целое число в диапазоне от 100 до 999. Составить программу, которая выводит текстовое описание данного числа, например: 145 – «сто сорок пять», 768 – «семьсот шестьдесят восемь».

 Пригласите преподавателя и продемонстрируйте ему все выполненные вами задания. После этого сохраните все программы и выйдите из среды *PascalABC.NET*. Будьте готовы ответить на вопросы преподавателя.

 Если вы обучаетесь в условиях использования дистанционных образовательных технологий, просто вышлите все файлы с подготовленными вами программами преподавателю.

## Глава 8

### ЦИКЛ С ПАРАМЕТРОМ

#### Задание 56

В электронном пособии (Examles\Пособие\index.html) изучите раздел «Введение в систему типов языка Паскаль». Ответьте на вопросы, приведенные в конце раздела. Выполните тест по теме и продемонстрируйте оценку преподавателю.

#### Задание 57

В электронном пособии (Examles\Пособие\index.html) изучите раздел «Циклы с параметром: быстрое начало».

Запустите среду PascalABC.Net и проверьте работу всех типовых алгоритмов, приведенных в разделе.

Ответьте на вопросы, приведенные в конце раздела. Выполните тест по теме и продемонстрируйте оценку преподавателю.

#### Задание 58

Считайте с диска файл с именем **lab8\_55.pas**. Содержащаяся в нем программа 15 раз печатает на экране слово «Халва...» (убедитесь в этом, запустив программу). Модифицируйте программу так, чтобы:

- а) слово печаталось не 15, а 10 раз;
- б) слова печатались в одну строку;
- в) перед первым словом печаталось слово «Начало», а после последнего – слово «Конец»;
- г) каждое слово печаталось с новой строки и между ними была пустая строка;
- д) перед каждым словом «Халва...» печатался его порядковый номер (значение переменной  $i$ ).

#### Задание 59

В файле **lab8\_56.pas** находится программа печати таблицы значений функции  **$f(x) = \sin(x)$**  при  $x = 0, 0.1, 0.2, \dots, 2$ . Модифицируйте программу так, чтобы:

- а) печатался заголовок;
- б) печатались еще и значения аргумента  $x$ ;

в) печатались значения при  $x = 0, 0.2, 0.4, \dots, 3$ ;

г) то же самое печаталось в обратном порядке.

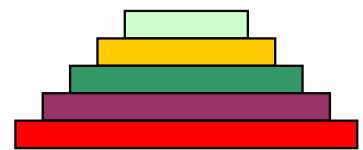
### Задание 60

В файле **lab8\_57.pas** находится программа движения окружности по горизонтали. Используя эту программу, смоделируйте на экране равномерное прямолинейное движение этой окружности:

- а) по вертикали;
- б) по диагонали окна.

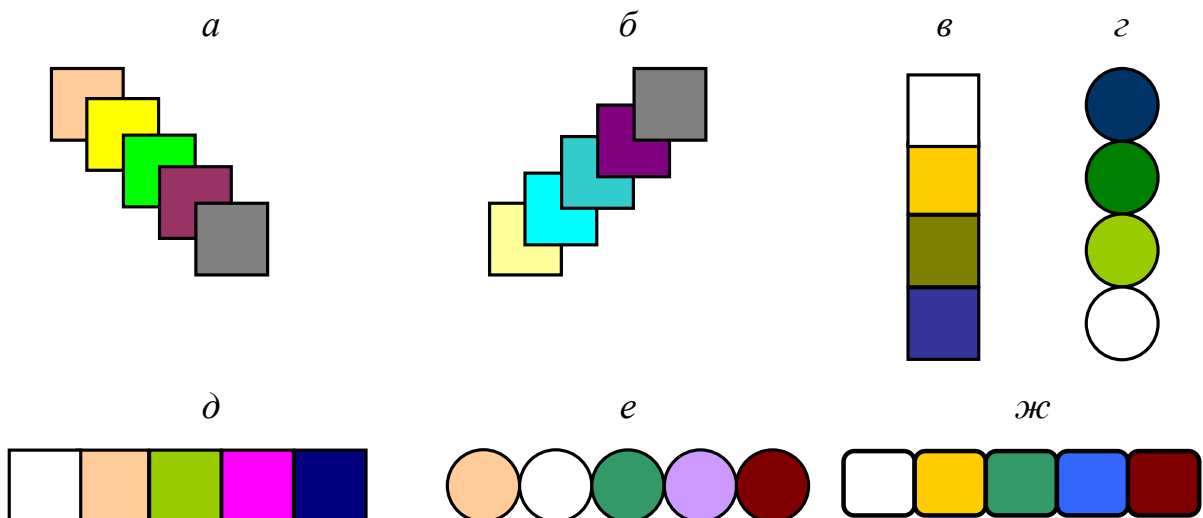
### Задание 61\*

Используя конструкцию цикла со счетчиком, самостоятельно составьте программу рисования детской игрушечной пирамидки (вид сбоку).

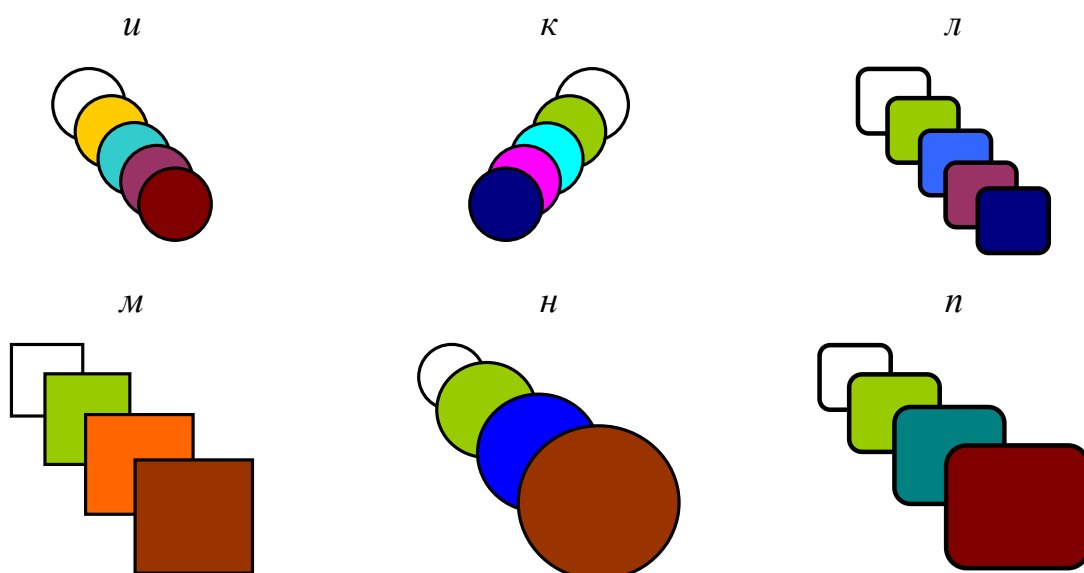


### Задачи для самостоятельной работы

1. Построить окружность, состоящую из 6 секторов, раскрашенных в разные цвета (для задания разных цветов можно использовать цвет **clRandom**).
2. Составить программу рисования мишени, состоящей из окружностей разного цвета.
3. Составить программу рисования следующей последовательности однотипных фигур:







☞ Пригласите преподавателя и продемонстрируйте ему все выполненные вами задания. После этого сохраните все программы и выйдите из среды *PascalABC.NET*. Будьте готовы ответить на вопросы преподавателя.

☞ Если вы обучаетесь в условиях использования дистанционных образовательных технологий, вышлите все файлы с подготовленными вами программами преподавателю.

## Глава 9\*

# РИСОВАНИЕ ЗАБОРА

Конечно, простейший забор можно нарисовать и с помощью прямоугольников. Но это не так интересно. Давайте попробуем нарисовать забор традиционный, из заостренных планочек. Как это сделать?

Познакомимся еще с несколькими процедурами библиотеки GraphABC. Для этого сначала поговорим о графическом курсоре.

В консольных приложениях курсор виден в виде мигающей черточки. В графическом же режиме курсора нет. Дело в том, что он указывает на конкретную маленькую точку, и если его нарисовать, то самой точки видно не будет. Поэтому введено понятие графического пера, под которым понимают некоторое невидимое перо, с помощью которого можно рисовать в графическом окне.

Кроме того, введено понятие кисти, т. е. некая невидимая кисть, которая позволяет закрашивать заданные области. Технология достаточно простая: если сначала задать цвет кисти (вы уже ею пользовались), а затем нарисовать, например, прямоугольную рамку с помощью процедуры **Rectangle**, то она будет закрашена внутри цветом кисти.

*Процедура установки графического пера:*

**MoveTo**(X, Y : **integer**); – передвигает перо к точке с координатами (x, y);

*Процедуры рисования линий:*

**LineTo**(X, Y : **integer**); – процедура позволяет проводить прямую из точки с текущими координатами в точку (x, y) текущим цветом пера;

**Line**(x1,y1,x2,y2 : **integer**); – процедура обеспечивает рисование отрезка из точки (x1, y1) в точку (x2, y2) текущим цветом пера.

*Пример:*

```
...  
MoveTo(10,10);  
LineTo(100,100); { проводит прямую в точку 110,110 }  
...
```

*Процедуры задания цвета и стиля линий:*

**SetPenColor**(color: **integer**); – процедура позволяет задать цвет пера. Перечень допустимых цветов для процедур и функций библиотеки GraphABC

см. в Справке (Помощь): Стандартные модули → Модуль GraphABC → GraphABC: цветовые константы.

**SetPenStyle(ps: integer);** – устанавливает стиль пера, задаваемый параметром ps.

Стиль пера (ps) определяется одной из предопределенных констант. Идентификаторы этих констант и определяемые ими стили см. в Справке (Помощь): Стандартные модули → Модуль GraphABC → GraphABC: стили пера;

**SetPenWidth(w: integer);** – устанавливает ширину пера, равную w пикселям;

**SetBrushColor(color: integer);** – устанавливает цвет кисти, задаваемый параметром **color**. Используются только цвета библиотеки GraphABC.

*Пример:*

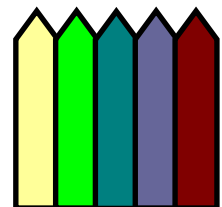
```
uses GraphABC;
begin
  SetPenColor(clFuchsia);    {цвет пера – для линии}
  SetPenWidth(10);           {толщина пера – для линии}
  SetBrushColor(ClBlue);     {задаем цвет кисти – закразка}
  SetBrushStyle(psDash);     {тип кисти – закразка }
  Rectangle(10, 10, 180, 100); {рисуем закрашенный прямоугольник }
  SetBrushStyle(bsSolid);
  FloodFill(210, 110, clIndianRed); {закрашиваем область вне прямоугольника }
end.
```

☞ Чтобы не использовать идентификаторы цветов, используйте случайный цвет. Для этого вместо константы, обозначающей цвет, укажите идентификатор *clRandom* (т. е. случайный цвет).

## Задание 62

Запустите среду PascalABC.NET.

Теперь давайте рисовать забор, состоящий из пятиугольных досок (см. рисунок). При рисовании забора не забудьте раскрасить каждую доску в свой цвет.



При рисовании линий используйте **LineTo**.

Успехов вам!

## Задание 63

Вы нарисовали забор. Модифицируйте программу таким образом, чтобы был нарисован забор длиной:

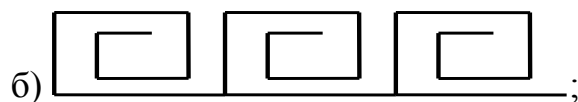
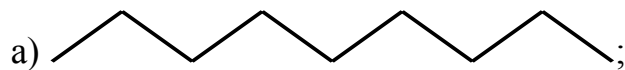
- а) в 500 точек (пикселей) из 10 досок;
- б) в 400 пикселей из заданного числа досок (количество досок вводится с клавиатуры);
- в) в  $N$  пикселей и из  $M$  досок.

### Задание 64

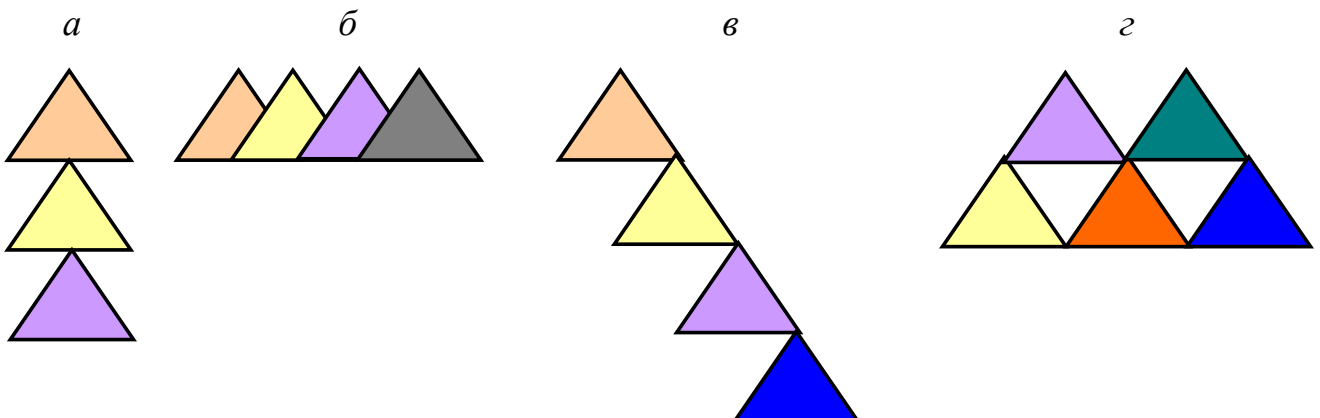
Еще одна модификация: измените программу так, чтобы был нарисован забор длиной в  $N$  пикселей, причем длина одной доски равнялась  $K$  пикселей.

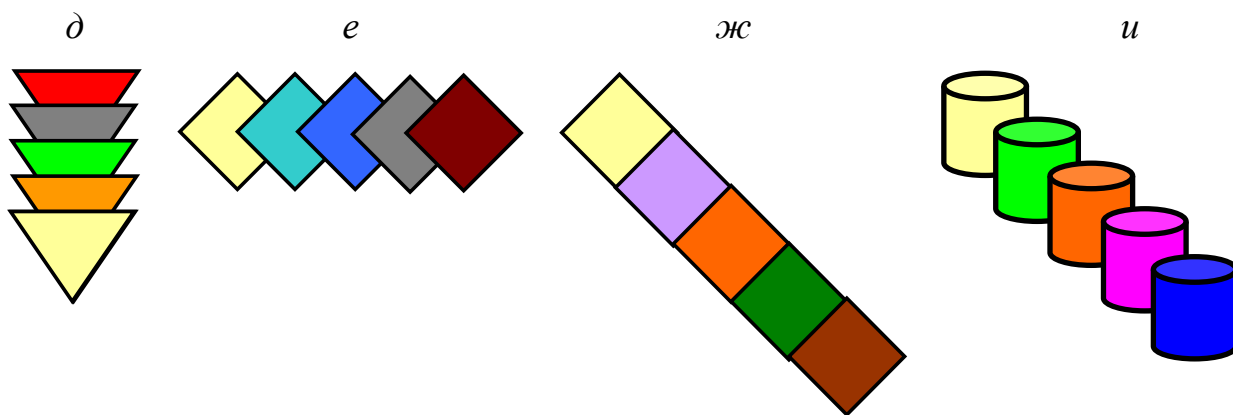
### Задачи для самостоятельной работы

1. Составьте программу рисования следующего орнамента:



2. Составьте программу (с использованием цикла) рисования правильного шестиугольника.
3. Составьте программу рисования пятиугольной звездочки.
4. Составьте программу рисования следующей последовательности однотипных фигур:





- ☞ *Пригласите преподавателя и продемонстрируйте ему все выполненные вами задания. После этого сохраните все программы и выйдите из среды PascalABC.NET. Будьте готовы ответить на вопросы преподавателя.*
- ☞ *Если вы обучаетесь в условиях использования дистанционных образовательных технологий, вышлите все файлы с подготовленными вами программами преподавателю.*

## Глава 10<sup>0</sup>

# ЗВЕЗДНОЕ НЕБО И ПРОЧИЕ СТРАННОСТИ

### 1. Получение случайного числа

При создании некоторых программ (например, игр) требуется получение случайного числа при каждом запуске программы (например, если вы моделируете на компьютере игру в кости, карты и т. п.). Для этих целей создана специальная *функция*

**Random(*n* : integer) : integer; .**

Эта функция выдает случайное целое число в промежутке от 0 до  $n-1$ , где  $n$  – заданное целое положительное число.

Кроме того, имеется аналогичная функция без параметров:

**Random: Real;**

Данная функция выдает случайное вещественное число в промежутке от 0 до 1.

Посмотрим, как это работает.

#### Задание 65

Запустите среду **PascalABC.NET**.

Считайте программу из файла **rnd\_62.pas** и разберитесь, как она работает и что выдает на экран дисплея. Для этого запустите эту программу 2–3 раза и определите, какую последовательность чисел она выдает: одну и ту же при каждом запуске или различные.

#### Задание 66

Считайте программу из файла **rnd\_63.pas** и разберитесь, как она работает и что выдает на экран дисплея. Для этого запустите эту программу 2–3 раза и определите, какую последовательность чисел она выдает: одну и ту же при каждом запуске или различные. Определите, в чем отличие этих программ.

#### Задание 67

Считайте программу из файла **rnd\_64a.pas** и разберитесь, как она работает. Выясните, как там организован цикл и когда он завершает свою работу.

Считайте программу из файла **rnd\_646.pas** и попытайтесь разобраться, как она работает. Выясните, как там организован цикл и когда программа завершает свою работу.

### Задание 68

Используя алгоритм, предложенный в программе **rnd\_646.pas**, составьте программу, которая выводит на экран дисплея закрашенные прямоугольники в произвольном месте экрана произвольных размеров и произвольного цвета. Программа должна работать бесконечно (т. е. до тех пор, пока не будет закрыто окно). Очищать окно после каждого вывода не нужно.

## 2. Рисование точек в графическом режиме

Точки в языке Паскаль рисуются достаточно просто: достаточно воспользоваться процедурой

**PutPixel(X, Y, Color: integer);**

Эта процедура позволяет закрашивать на **экране** точку с координатами *X* и *Y* цветом **Color** (**integer** – это описание типа параметров процедуры). Например, если вам хочется нарисовать точку голубого цвета с координатами (100,200), то потребуется в программе лишь обратиться к процедуре вывода точки следующего вида:

**PutPixel(100,200,clblue);**

### Задание 69

Составьте программу «Звездное небо», которая постоянно выводит точки разного цвета в разных местах окна. Программа должна работать бесконечно (т. е. до тех пор, пока не будет закрыто окно).

### Задание 70

Откройте файл **rnd\_67.pas**. Разберитесь в тексте программы. Измените программу так, чтобы она рисовала более крупные звезды и число лучей в них изменялось от 3 до 8.

## Задачи для самостоятельной работы

Учебные тренажеры работают следующим образом:

- выдается некоторое случайным образом подобранное задание;
- пользователь вводит ответ;

- если ответ верный, обучаемому выдается следующее задание;
- если ответ неверный, то обучаемому выдается верный ответ и сообщение о количестве допущенных им ошибок.

Для создания заданий используйте датчик случайных чисел.

1. Разработайте простейший тренажер для учащихся младших классов, который проверяет:
  - а) знание таблицы умножения (т. е. умение перемножать числа от 1 до 10);
  - б) умение складывать числа от 1 до 100;
  - в) навык вычитания чисел от 1 до 100. Следует иметь в виду, что учащиеся начальной школы еще не знают отрицательных чисел, поэтому уменьшаемое должно быть больше вычитаемого;
  - г) умение делить целые числа до 100. Здесь должны проверяться действия, обратные умножению целых чисел т. е. деление должно быть целочисленное (делимое должно делиться делителем нацело).
2. Разработайте простейший тренажер, который проверяет умение решать:
  - а) линейные уравнения вида  $ax = b$ ;
  - б) квадратные уравнения вида  $ax^2 + bx + c = 0$ .

☞ *Пригласите преподавателя и продемонстрируйте ему все выполненные вами задания. После этого сохраните все программы в своей папке и выйдите из среды PascalABC.NET. Будьте готовы ответить на вопросы преподавателя.*

☞ *Если вы обучаетесь в условиях использования дистанционных образовательных технологий, вышлите все файлы с подготовленными вами программами преподавателю.*



## Глава 11

# ЦИКЛ С ПАРАМЕТРОМ: ТИПОВЫЕ АЛГОРИТМЫ

### Задание 71

В электронном пособии ([Examples\Пособие\index.html](#)) изучите раздел «Конструкция цикла FOR». Обратите внимание на типовые алгоритмы. Ответьте на вопросы, приведенные в конце раздела. Выполните тест по теме и продемонстрируйте оценку преподавателю.

## 1. Вычисление суммы

### Задание 72

Запустите среду **PascalABC.NET**. Выполните следующие действия:

- а) загрузите файл **lab11\_68.pas**;
- б) прочитайте текст программы и проанализируйте его;
- в) запустите программу на выполнение (рекомендуем задать в качестве значения числа  $N$  значения 1, 10, 0, -5); оцените, как программа ведет себя в различных ситуациях);
- г) измените программу так, чтобы она считала сумму кубов первых  $N$  чисел.

### Задание 73

Выполните следующие действия:

- а) загрузите файл **lab11\_69.pas**;
- б) прочитайте текст программы и проанализируйте его;
- в) запустите программу на выполнение;
- г) модифицируйте программу так, чтобы она выводила таблицу Пифагора 15 на 15.

## 2. Вычисление факториала

### Задание 74

Внимательно просмотрите перечисленные ниже программы, поскольку на их основе потребуется самостоятельно реализовывать аналогичные алгоритмы:

- а) lab11\_70a.pas – если значение факториала записывать в переменную типа `integer`, то быстро наступает переполнение. Выясните, при каком значении введенного числа это переполнение наступает (т. е. в качестве ответа выводится отрицательное число или 0);
- б) измените программу вычисления значения факториала, заменив тип `integer` на `int64`. Выясните, при каком значении введенного числа наступает переполнение с этим типом данных;
- в) lab11\_70в.pas – предлагается еще один вариант решения задачи: результат записывать в переменную типа `real`. Выясните, при каком значении введенного числа переполнение наступает в этом случае.

## 3. Вычисление суммы ряда

### Задание 75

Квадрат любого натурального числа  $N$  равен сумме первых  $N$  нечетных чисел:

$$1^2 = 1$$

$$2^2 = 1 + 3$$

$$3^2 = 1 + 3 + 5$$

$$4^2 = 1 + 3 + 5 + 7 .$$

Проверьте, верна ли эта закономерность для других натуральных чисел.

### Задание 76

Многие из математических величин или значений функций могут быть выражены как суммы бесконечных последовательностей. Чем больше членов ряда участвует в сложении, тем более точным получается искомое значение.

Составьте программы и попытайтесь определить минимальное число членов ряда, обеспечивающих вычисление следующих данных:

а)  $e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \dots + \frac{1}{n!} + \dots$ ; (значение  $e = 2,718\ 281$ );

б)  $\pi = 4(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots + \frac{(-1)^{n-1}}{2n-1} + \dots)$  (значение  $\pi = 3,141\ 592$ );

в)  $\sqrt{2} = 2 \cdot \left(1 - \frac{1}{3}\right) \cdot \left(1 + \frac{1}{5}\right) \cdot \left(1 - \frac{1}{7}\right) \cdot \dots \cdot \left(1 - \frac{(-1)^k}{2k+1}\right)$  (значение  $\sqrt{2} \approx 1,414\ 213$ )

#### 4. Определение среднего

##### Задание 77

Для определения рентабельности авиаинии вычисляется среднее количество пассажиров за год. Составьте программу вычисления среднего количества пассажиров, перевезенных за год, если известно общее количество пассажиров, перевезенных за каждый месяц.

##### Задание 78

Составьте программу определения среднемесячной температуры воздуха, если известна средняя температура за каждый день месяца. Исходные данные для вычислений лучше использовать реальные.

#### 5. Определение экстремума

##### Задание 79

В память компьютера вводят по очереди координаты  $N$  точек. Определить координаты точки, максимально удаленной от начала координат.

##### Задание 80

У тренера футбольного клуба имеется список членов команды и количества очков, которые принес команде каждый игрок в данном сезоне. Составить программу, с помощью которой можно определить самого результативного игрока.

## 6. Рекуррентные соотношения

В 1202 году итальянский математик Леонард Пизанский, известный под именем Фибоначчи, предложил такую задачу:

Пара кроликов каждый месяц дает приплод – пару кроликов (самца и самку), от которых через два месяца уже получается новый приплод. Сколько кроликов будет через год, если в начале года мы имели одну пару молодых кроликов?

Обратим внимание на то, что числа, соответствующие количеству пар кроликов, которые имеются через каждый месяц, составляют последовательность:

$$1, 1, 2, 3, 5, 8, 13, 21, 34, \dots$$

Каждый из членов этой последовательности, начиная с третьего, равен сумме двух предыдущих членов. Эта последовательность получила название ряда Фибоначчи, а ее члены – числа Фибоначчи.

Обозначив  $n$ -й член ряда Фибоначчи  $F(n)$ , мы получим следующую зависимость, называемую *рекуррентной* (т. е. зависимость, в которой последующие элементы определяются предыдущими):

$$F(1)=1$$

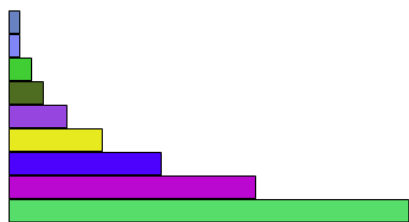
$$F(2)=1$$

$$F(n)=F(n-1)+F(n-2), \text{ при } n \geq 3$$

### Задание 81

Составить программу, которая:

- а) выводит первые  $N$  чисел Фибоначчи;
- б) определяет, сколько кроликов будет через  $M$  месяцев; через  $G$  лет;
- в) позволит определить номер члена последовательности, значение которого превосходит заданное число  $K$ ;
- г) представить ряд Фибоначчи графически. Например:



(длина прямоугольника пропорциональна числу; например, его длина равна числу, умноженному на 10).

### Задачи для самостоятельной работы

1. Напишите программу вычисления наибольшего общего делителя двух натуральных чисел.
2. Оборудование фирмы в результате износа и старения уценивается на  $p\%$  ежегодно. Составьте программу, которая по первоначальной стоимости оборудования и времени его эксплуатации вычисляет текущую стоимость этого оборудования.
3. Напишите программу, которая проверяет, является ли простым введенное пользователем положительное целое число. Простым называется такое число, которое делится без остатка только на единицу и само себя (например, 5, 11, 19).
4. Напишите программу вычисления совершенных чисел, не превосходящих заданного числа  $N$ . Совершенным называется такое число, сумма делителей которого совпадает с самим числом (например,  $6 = 1 + 2 + 3$ ).
5. Для каждого посетителя парикмахерской (с одним мастером) известны следующие величины:  $t$  – момент его прихода;  $\tau$  – продолжительность его обслуживания. Сколько клиентов обслужит мастер за смену продолжительностью  $T$ ? Сколько рабочего времени он потратит на обслуживание?
6. Расписание звонков. В учебном заведении задается начало учебного дня, продолжительность пары или урока, продолжительность обычного и большого перерывов (и их место в расписании), количество пар (уроков). Составить программу, которая выведет на экран расписание звонков на весь учебный день.
7. Гуси и кролики. У гусей и кроликов  $2n$  лап. Сколько может быть гусей и кроликов (вывести все возможные сочетания)?
8. Составить программу, которая позволит определить, сколько можно купить быков, коров и телят, платя за быка 10 р., за корову – 5 р., а за теленка – 0,5 р., если на 100 р. надо купить 100 голов скота?

9. Составьте программу, которая выводит на экран календарь на любой месяц по заданному году. Воспользуйтесь алгоритмом вычисления дня недели, описанном в гл. 5.

☞ *Пригласите преподавателя и продемонстрируйте ему все выполненные вами задания. После этого сохраните все программы в своей папке и выйдите из среды PascalABC.NET. Будьте готовы ответить на вопросы преподавателя.*

☞ *Если вы обучаетесь в условиях использования дистанционных образовательных технологий, вышлите все файлы с подготовленными вами программами преподавателю.*

## Глава 12

# ПРОЕКТИРОВАНИЕ ПРОГРАММ И ПРОЦЕДУРЫ

### 1. Необходимость технологии программирования

Предположим, вам захотелось построить дом. Возможно два варианта:

- «любительский» – берем кирпич (дерево, плиту, бетон...) и строим. Что-нибудь получится. Если не развалится. А если развалится – будем пробовать еще. Пока не получится;

- «профессиональный» – создаем сами или заказываем ЭСКИЗ дома (фасад, количество комнат, подземных гаражей, туалетов, размеры и количество окон и дверей и т.д.), затем – ПРОЕКТ (подробнейший план дома с размерами деталей и структурой материала), далее – привязка дома к ландшафту (чтобы было удобно строить и красиво смотрелось) и только потом уж начинается практическая РЕАЛИЗАЦИЯ: фундамент, стены, перекрытия, крыша, системы обеспечения (отопление, водопровод, канализация, электричество, телефон), внутренняя отделка, строительство газонов, установка мебели.

Цепочка ЭСКИЗ → ПРОЕКТ → РЕАЛИЗАЦИЯ имеет отношение практически ко всем видам производственной деятельности: создание машин, картин, одежды, и в том числе программ для компьютеров. Когда речь идет о программе от 5 до 50 строк, можно воспользоваться «любительским» вариантом: сел и пиши. Если же речь идет о быстром написании больших программ, необходима специальная технология, обеспечивающая как скорость, так и качество программного продукта. Одной из таких технологий является технология программирования «сверху вниз». С примером разработки программы по такой технологии вы и познакомитесь в данной лабораторной работе.

### 2. Знакомство с технологией проектирования «сверху вниз»

Выполним следующее задание.

*Составить программу, которая рисует на экране дисплея еловый лес.*

## РЕШЕНИЕ

### 1. Уточнение постановки задачи (ЭСКИЗ)

Выясним подробнее, что же все-таки нам нужно нарисовать. Договоримся, что под «еловым лесом» будем понимать набор елочек различных размеров и различной расцветки, расположенных в произвольных местах экрана.

### 2. Первый вариант решения

Предположим, что у нас есть уже готовая процедура **Forest(n)** (n – количество деревьев в лесу), при вызове которой рисуется требуемый лес. Тогда наша программа выглядит достаточно просто:

```
uses GraphABC;  
begin  
    Forest(25); // Вызов процедуры – рисуем лес из 25 деревьев  
end.
```

К сожалению, если эту программу запустить в **PascalABC.NET**, будет выдана ошибка, означающая, что среда не знает процедуры **Forest**.

### 3. Уточнение решения

Следовательно, такую процедуру надо создать. Сначала познакомимся с тем, как это делается.

## ТЕОРИЯ

Вы уже знакомы со стандартными процедурами типа **Rectangle(x,y,x1,y1)**, **SetPenColor(m)** и т. п. Язык программирования Паскаль позволяет вам создавать свои собственные процедуры и функции.

 **Процедура** представляет собой часть программы, оформленной специальным образом.

Любая процедура имеет структуру, напоминающую структуру программы:

```
Procedure <имя >[(<список формальных параметров>);]  
    <блок описания данных>  
begin  
    <операторы>  
end;
```



Обратите внимание на структуру заголовка процедуры и точку с запятой в конце процедуры – именно этим оформление процедур отличается от оформления программ.

Процедуры, как и переменные, должны быть описаны непосредственно в программе, в которой они используются. Описание процедур в программе располагается после блока описания переменных VAR. В этом случае программа имеет следующую структуру:

```
Program <имя программы>; // необязательная строка
<блок описания констант>
<блок описания типов>
<блок описания переменных>
<блок описания процедур>
begin
    <операторы>
end.
```

#### Задание 82

1. Запустите среду PascalABC.NET. Загрузите программу **Proc1.pas**. Разберитесь в тексте программы. Обратите внимание на оформление заголовка и расположение процедуры в тексте программы.
2. Запустите программу на выполнение и выясните, что она делает.
3. Модифицируйте программу так, чтобы она рисовала забор толщиной 15 и высотой 75 пикселей, а доски располагала ступеньками сверху вниз по диагонали экрана.

#### ТЕОРИЯ

Заголовок процедуры включает в себя ключевое слово PROCEDURE (читается «просидже» и переводится как «процедура» или «образ жизни»), имя процедуры и описание параметров, т. е. тех данных, которые передаются в процедуру извне и используются в теле процедуры. Типы параметров описываются непосредственно в заголовке.

Различные параметры с описанием типов отделяются друг от друга точкой с запятой. В процедуре может и не быть параметров.

Например:

Procedure **House**(**x,y**:word; **name**:string; **usd**:real); – процедура с 4 параметрами (2 целых числа, строка и вещественное число).

...

Procedure **Man**(**name,adress**:string); – заголовок процедуры с 2 параметрами (строками).

...

Procedure **Girl**(**name**:string; **eye**:byte; **nice**:real); – процедура с 3 параметрами (строка, целое число от 0 до 255, вещественное число).

...

Procedure **Wait**; – заголовок процедуры без параметров.

...

### Задание 83

1. Загрузите программу **Proc2.pas**. Разберитесь в тексте программы. Выясните последовательность описания вызываемой и вызывающей процедур.
2. Переделайте программу таким образом, чтобы сам фрагмент рисования трех заборов был выделен в отдельную процедуру **pict** БЕЗ ПАРАМЕТРОВ. При этом тело программы должно выглядеть следующим образом:

```
...  
begin  
    PICT // Обращение к процедуре рисования трех заборов  
end.
```

### Задание 84

Загрузите программу из файла **Proc3.pas**. Допишите в ней процедуру **TRIANGLE**, рисующую равносторонний закрашенный треугольник.

☞ *Сохраните все программы, пригласите преподавателя и продемонстрируйте их ему. В случае, если вы обучаетесь дистанционно, после выполнения всей лабораторной работы тексты программ отправьте тьютору.*

### 4. Уточнение решения

Теперь вы знаете, что в программе рисования леса следует описать процедуру **Forest**.

### Задание 85

Загрузите файл **Proc4.pas**. В этом файле находится текст уточненной программы. Запустите программу. Убедитесь в том, что программа не имеет

ошибок и исполняется. Но программа ничего не делает, поскольку ее следует еще дописать, вставив вместо комментариев реальную процедуру (или процедуры).

#### 5. Уточнение решения

Давайте предположим, что у нас есть еще одна графическая процедура **Fir** (по-английски «ель»), которая позволяет нарисовать елочку. Пусть данная процедура имеет следующий заголовок:

```
procedure fir(x,y,h:integer; c1,c2:Color);
```

где x,y – координаты елочки; h – высота елочки; c1 – цвет линий; c2 – цвет закрашки. Под «координатами елочки» договоримся понимать, например, координаты ее вершины. Тогда программа будет выглядеть следующим образом.

**Uses GraphABC;**

---

```
procedure triangle(x,y,h:integer; c1,c2: color); // Рисование треугольника
{Параметры: x,y – координаты вершины; h – высота;
 c1 – цвет линий; c2 – цвет закрашки }
begin
    // Рисуем треугольник
end;
```

---

```
procedure fir(x,y,h:integer; c1,c2: color); // Процедура рисования елочки
{Параметры: x,y – координаты вершины; h – высота; c – цвет линий;
 c2 – цвет закрашки }
begin
    // Рисуем елочку, обращаясь к процедуре TRIANGLE
end;
```

---

```
                                // процедура рисования леса
procedure Forest (N:integer);    // N – количество елочек в лесу
const h0=30;                    // высота треугольника
var i: integer;
    a,b,h: integer;
    color1,color2,: color;
begin
    For i:=1 to N do
    begin
        h:=random(h0)+10;        // высота треугольника
        a:=random(WindowWidth)+20; // координата x вершины елочки
        b:=random(WindowHeight)+5; // координата y вершины елочки
        color1:=ClrRandom;       // цвет линий
```

```

    color2:=CIRandom;           // цвет закраски
    fir(a,b,h,color1,color2);  // рисуем елочку
end
end;

```

---

```

begin          { тело программы }

```

```

    Forest(25);

```

```

end.

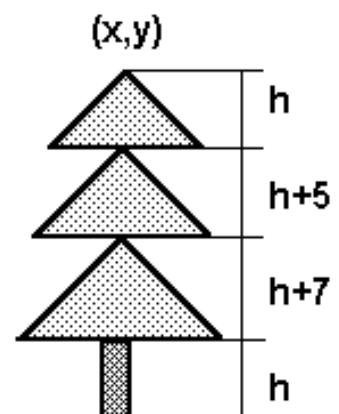
```

Приведенная выше программа также не имеет ошибок, выполняется, но все же леса по-прежнему не рисует.

## 6. Уточнение решения

Теперь попробуем понять, а как, собственно, рисует процедура **Fir**. Для простоты пусть елочка состоит из 3 равнобедренных треугольников и ножки (см. рисунок).

Под «высотой» елочки проще всего считать высоту верхнего треугольника. Цвет елочки – цвет треугольника. Высоту ножки елочки можно задать, например, равной высоте первого треугольника. Треугольники, расположенные ниже, могут иметь высоту немного побольше (например,  $h+5$  и  $h+7$  соответственно). Вариант размеров елочки вы можете посмотреть на рисунке. Понятно, что эти размеры условны и, если вам хочется, вы можете выбрать их другими.



Таким образом, процедура **Fir** рисует елочку из 3 закрашенных треугольников и ножки.

## Задание 86

Учитывая, что процедура рисования треугольника у вас уже написана (не забудьте ее использовать в проекте), завершите реализацию программы рисования леса. Для ускорения работы воспользуйтесь уже готовой частью программы из файла **Proc4.pas**.

При рисовании вы, несомненно, заметили дефекты закрашки треугольников. Эту ситуацию можно исправить, если воспользоваться процедурой рисования многоугольников **polygon**.



### Задание 87

Откройте файл **ромбы.pas**. Разберитесь в технологии рисования ромбов.

Модифицируйте программу так, чтобы она рисовала треугольники.

На основе этой программы составьте процедуру рисования треугольника и замените ею процедуру **triangle**. Все проблемы будут устранены.

### Задание 88

Модифицируйте программу таким образом, чтобы вместе с лесом рисовались домики () или машинки (). Количество фигурок должно соответствовать числу елей. В программе должна быть отдельная процедура рисования выбранного объекта.

*Это вам пригодится*

Обратите внимание на этапы деятельности при проектировании программы рисования леса – это очень важно! Дело в том, что практическое программирование – это на 70 % проектирование программы и только на 30 % – запись программы на языке программирования.

Описанный данный вариант проектирования разработан и используется в наиболее известных программистских фирмах мира. По этой технологии (напомним, что называется она «технология программирования “сверху вниз”»), в частности, разработана и реализована сама среда PascalABC.NET. Такая технология позволяет достаточно быстро разрабатывать сколь угодно большие программы.

Быстрота разработки обеспечивается тем, что после создания проекта программы (на уровне простых процедур) реализация процедур может быть поручена нескольким программистам, и, если в проекте нет серьезных ошибок, программа будет готова тогда, когда последний программист закончит писать свою процедуру. Далее, правда, следует этап отладки программы, но это предмет другой лабораторной работы.

☞ *Пригласите преподавателя и продемонстрируйте созданный вами проект.*

☞ *При обучении в условиях дистанционного обучения отправьте проект по электронной почте.*

## Глава 13

# ЦИКЛЫ С УСЛОВИЕМ

### Задание 89

В электронном пособии (Examples\Пособие\index.html) изучите раздел «Циклы с условием». Ответьте на вопросы, приведенные в конце раздела. Выполните тест по теме и продемонстрируйте оценку преподавателю.

### 1. Цикл с предусловием (WHILE)

Цикл со счетчиком используется тогда, когда количество повторений цикла заранее известно.

Цикл WHILE используется тогда, когда количество повторений заранее неизвестно, но известно *условие завершения* выполнения цикла. При этом условие должно проверяться *до выполнения* оператора в цикле.

Общая структура цикла:

**WHILE** <условие> **DO** <оператор>

Цикл выполняется до тех пор, пока <условие> **TRUE** (ИСТИННО).

### Задание 90

Выполните следующие действия:

- а) загрузите файл **lab13\_86.pas**;
- б) прочитайте текст программы и проанализируйте его;
- в) запустите программу на выполнение (рекомендуем задать значения 1000, 200, 0, -5 в качестве значения M). Оцените, как программа ведет себя в различных ситуациях;
- г) измените программу так, чтобы она проверяла, является ли введенное число суммой кубов первых  $N$  натуральных чисел. Если является, следует указать это  $N$ . Для проверки используйте число 2025, здесь  $N = 9$ .

## 2. Цикл с постусловием (REPEAT)

Этот цикл используется также в ситуации, когда количество повторений неизвестно, а известно условие завершения работы цикла. В таком случае само условие должно проверяться после выполнения действий в цикле.

Общая структура цикла:

**REPEAT**

<оператор 1>;

<оператор 2>;

...

<оператор N>

**UNTIL <условие>**

Цикл выполняется до тех пор, пока <условие> **FALSE** (ЛОЖНО).

☞ Обратите внимание, что цикл с предусловием, в отличие от других операторов языка, может содержать любое количество операторов!

### Задание 91

Выполните следующие действия:

- а) загрузите файл **lab13\_87.pas**;
- б) прочитайте текст программы и проанализируйте его;
- в) запустите программу на выполнение;
- г) измените программу так, чтобы она завершала работу при  $A = 999$ .

### Задание 92

Выполните следующие действия:

- а) загрузите файл **lab13\_88.pas**;
- б) прочитайте текст программы и проанализируйте его;
- в) запустите программу на выполнение;
- д) измените программу так, чтобы в гонке участвовал еще третий участник.

### Задание 93

Составьте программу, которая вычисляет сумму ряда:

а)  $\pi = 4 \cdot \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots + \frac{(-1)^{n-1}}{2n-1} + \dots\right)$  (значение  $\pi \approx 3,141\,592$ );

$$\text{б) } e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \dots + \frac{1}{n!} + \dots \text{ (значение } e \approx 2,718\,281\text{)}.$$

Добавлять члены последовательности необходимо до тех пор, пока очередной член не станет меньше заданной точности  $\varepsilon$ .

### Задачи для самостоятельной работы

1. В компьютер автомастерской после обслуживания клиента вводится стоимость ремонта. В конце дня вместо стоимости вводится число 0, после чего компьютер выводит на экран дисплея общую сумму выручки за день. Составьте такую программу.
2. Напишите программу, которая выводит двоичное представление введенного с клавиатуры десятичного числа.
3. Напишите программу, которая выводит шестнадцатеричное представление введенного с клавиатуры двоичного числа.
4. Напишите программу, которая выводит десятичное представление введенного с клавиатуры двоичного числа.
5. Напишите программу рисования шахматной доски.
6. Напишите программу-игру «Угадай число». Один игрок, в роли которого выступает компьютер, задумывает число в заданных пределах (например, от 1 до 999), другой игрок – человек – пытается его отгадать за минимальное число попыток. Для каждого числа, введенного человеком в качестве очередной версии, компьютер выдает сообщение «мое число больше», «мое число меньше» или «поздравляю, вы выиграли!», если число угадано точно. Предусмотрите начисление очков обратно пропорционально количеству затраченных попыток.
7. Дано натуральное число. Определить сумму его цифр.
8. В одной старинной задаче мудрец предложил шаху в качестве награды за услугу выдать ему зерно. Причем количество зерна оговорил так: на первую клетку шахматной доски положить 1 зерно, на вторую – 2, на третью – 4, на четвертую – 8 и т. д. Составить программу, с помощью



которой можно вычислить, на какой клетке суммарное количество зерен превысит  $N$  штук.

9. В компьютер по очереди вводятся баллы, полученные участницей на конкурсе красоты «Мисс Урал», а также номер участницы. Составить программу, которая обеспечит этот ввод и вывод на экран номер участницы, имеющей лучший результат.

☞ *Пригласите преподавателя и продемонстрируйте ему все выполненные вами задачи и задания. После этого сохраните все программы в своей папке и выйдите из среды PascalABC.NET. Будьте готовы ответить на вопросы преподавателя.*

☞ *Если вы обучаетесь в условиях использования дистанционных образовательных технологий, вышлите все файлы с подготовленными вами программами преподавателю.*

## Глава 14

# ПОСТРОЕНИЕ ГРАФИКОВ ФУНКЦИЙ

### Задание 94

В электронном пособии ([Examples\Пособие\index.html](#)) изучите раздел «Построение графиков». Ответьте на вопросы, приведенные в конце раздела. Выполните тест по теме и продемонстрируйте оценку преподавателю.

### Задание 95

Используйте описанную в теоретической работе программу (файл **lab14\_90.pas**) для построения графика функции  $f(x) = 2x + 30$ . Убедитесь, что программа работает правильно (из курса алгебры вам известно, что графиком такой функции является прямая линия, пересекающая оси  $Ox$  и  $Oy$  в точках  $-15,0$  и  $0,30$  соответственно).

### Задание 96

Пользуясь своей программой, постройте графики функций:

а)  $f(x) = |2x + 30|$ ;

б)  $f(x) = |||x - 80| - 40| - 20| - 10|$

(вместо знака модуля используйте функцию **abs**).

### Задание 97

Попробуйте с помощью той же программы построить график функции  $f(x) = 0,01x^2 + 3x - 7$ . Удалось ли это вам? Если нет, давайте разберемся, что произошло. А дело вот в чем: координаты точки на экране обязательно должны быть целыми числами, а значения функции  $f(x)$  оказываются в этом примере вещественными. Придется изменить описание переменных **x** и **y** на **Real**. Но это еще не все. Изменить нужно не только описание, но и вычисление значения переменной, округлив его до целого, поскольку координаты точки в процедуре **PutPixel** должны быть целыми. Для этого в языке Паскаль предусмотрена функция **Round** – округление. Используется она так:  $a := \text{Round}(b)$ , где  $a$  – переменная типа **Integer**;  $b$  – типа **Real**. Внесите изменения в программу и отладьте ее.

### Задание 98

Воспользуйтесь усовершенствованной программой для построения графика функции  $f(x) = \sin(x)$ . Похож ли полученный график на знакомую из курса алгебры синусоиду? Не очень, правда? Опять что-то не так. Действительно, значение функции  $\sin(x)$  лежит в пределах от -1 до +1, а график ее лежит совсем близко к оси  $Ox$ , почти сливаясь с ней. Когда вы рисуете график в тетради, вы интуитивно выбираете такой масштаб, чтобы график был хорошо виден. Компьютеру же надо дать точные указания о масштабе. Если умножить значение функции, например, на 50, то ординаты точек будут находиться в интервале шириной не 2, а 100 точек, и это уже будет хорошо видно на экране. Кроме того, чтобы масштаб по вертикали и по горизонтали был одинаковым, нужно при переходе от экранной координаты  $x$  к «реальной», поделить экранную координату на то же число (в данном случае 50), на которое умножили значение координаты  $y$ .

Если самостоятельно не удастся реализовать эту идею, разберитесь в программе **lab14\_93.pas**.

### Задание 99

Пользуясь рассмотренной в теоретической работе программой построения графика в полярной системе координат (файл **lab14\_94.pas**), постройте график функции  $r(a) = a$ . У вас должен получиться кусок спирали (часть графика может оказаться за пределами экрана).

Модифицируйте программу, чтобы нарисовать:

- а) окружность;
- б) спираль меньшего размера;
- в) несколько оборотов спирали;
- г) цветок, задаваемый в полярных координатах функцией  $r(a) = \sin(n \cdot a)$ , где параметр  $n$  задает число лепестков цветка. (Если ваш цветок будет слишком маленьким, воспользуйтесь идеей изменения масштаба, описанной в предыдущем задании.)

### Задание 100

Смоделируйте на экране дисплея движение шарика:

- а) по окружности;
- б) по спирали.

### Задание 101

Нарисуйте на экране несколько цветков, различных по окраске, размерам, количеству лепестков.

### Задачи для самостоятельной работы

1. Постройте следующие кривые, заданные в полярных координатах:

трехлепестковую розу:  $r = 2 \sin 2\varphi$ ;

циссоиду:  $r = \operatorname{tg} \varphi \sin \varphi$ ;

гиперболическую спираль:  $r = \frac{2}{\varphi}$ ;

строфоиду:  $r = \frac{1 + \sin \varphi}{\cos \varphi}$ ; ( $r = \frac{1 - \sin \varphi}{\cos \varphi}$ );

лемнискату Бернулли:  $r^2 = \cos 2\varphi$ ;

улитку Паскаля:  $r = a(1 + b \cos \varphi)$ ;

функцию  $r = \sqrt{\frac{\pi}{\varphi}}$ .

2. Постройте графики функций, заданных параметрически:

кардиоиду:  $x = a(t - \sin t)$ ;  $y = a(1 - \cos t)$ ;

астроиду:  $x = a \cos^3 t$ ;  $y = a \sin^3 t$ ;

декартов лист:  $x = \frac{3t}{1+t^3}$ ,  $y = \frac{3t^2}{1+t^3}$ .

3. Постройте следующие кривые, заданные в декартовой системе координат:

- $y = \frac{x^3 - 1}{4x^2}$ ;
- $y = (x^2 - x - 6)^2$ ;

- $|x| + |y| = 1;$
- $||x| + |y| - \frac{3}{2}| = \frac{1}{2}.$

☞ *Сохраните все программы в своей папке. Пригласите преподавателя и продемонстрируйте ему все выполненные вами задания. Будьте готовы ответить на вопросы преподавателя. После беседы с преподавателем выйдите из среды PascalABC.NET.*

☞ *Если вы обучаетесь в условиях использования дистанционных образовательных технологий, вышлите все файлы с подготовленными вами программами преподавателю.*

## Глава 15

### ГРАФИЧЕСКИЕ ПРИМИТИВЫ. ПЕРЬЯ И КИСТИ

Для дальнейшей работы желательно, чтобы вы познакомились еще с некоторыми процедурами, предоставляющими ряд дополнительных возможностей.

#### Задание 102

Запустите PascalABC.NET. Откройте Справку (пункт меню Помощь), раздел Стандартные модули → Модуль GraphABC → GraphABC: графические примитивы.

Прочитайте описание процедур Circle, FillCircle, DrawCircle, Ellipse, FilleEllipse, DrawEllipse, Arc, Pie, Chord, RoundRect, FillRoundRect, DrawRoundRect.

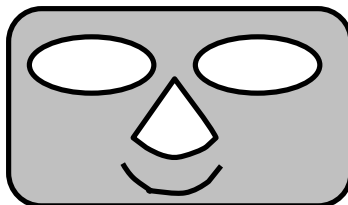
#### Задание 103

Нарисуйте окружность, состоящую:

- а) из двух закрашенных разными цветами полуокружностей;
- б) трех « « « секторов;
- в) четырех « « « секторов;
- г)  $n$  « « « секторов;

#### Задание 104

Составьте программу, рисующую следующее изображение:



Все элементы рисунка должны быть разного цвета, толщина линий не менее 3 точек, заливка цветная, но такая же полосатая.

#### Задание 105

Откройте файл SetPixel.pas. Просмотрите текст программы, выясните, что она делает.

Составьте программу, рисующую закрашенные сектора. Цвета, размеры и положения секторов определяются случайным образом.

#### **Задание 106**

Линии рисуются в PascalABC.NET с помощью пера (цвет, толщина и стиль линии совпадает с соответствующими характеристиками описания свойств пера). С помощью Справки познакомьтесь с технологией определения характеристик пера (Помощь – раздел Стандартные модули → Модуль GraphABC → GraphABC: стили пера).

#### **Задание 107**

Составьте программу, которая рисует 5 линий каждым из стилей пера. Все линии должны быть разного цвета. Попробуйте поменять толщину линий, посмотрите, сохраняются ли стили линий.

#### **Задание 108\*\***

Составьте программу штриховки наклонными пунктирными линиями:

- а) прямоугольника;
- б) эллипса;
- в) произвольного замкнутого контура.

#### **Задание 109**

Закраска замкнутых объектов (встроенных примитивов) реализуется в PascalABC.NET с помощью кисти (цвет и стиль совпадает с соответствующими характеристиками описания свойств кисти). С помощью Справки познакомьтесь с технологией определения характеристик кисти (Помощь – раздел Стандартные модули → Модуль GraphABC → GraphABC: стили кисти, GraphABC: текущая кисть Brush и GraphABC: подпрограммы работы с кистью).

#### **Задание 110**

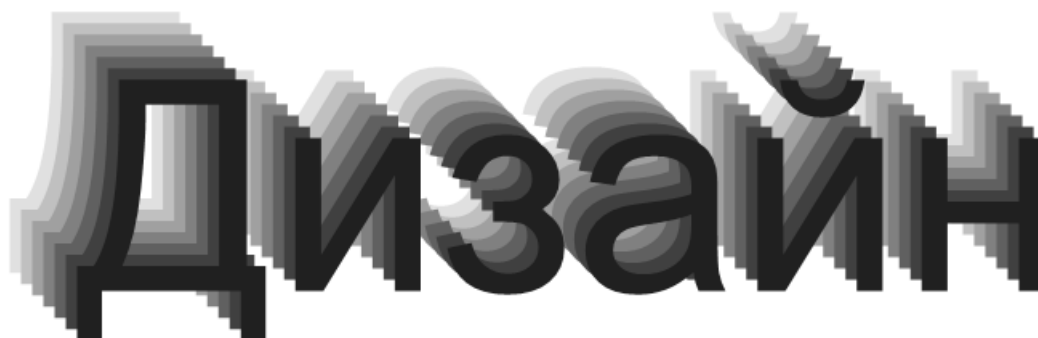
Составьте программу, которая рисует 5 прямоугольников, закрашенных каждым из допустимых для кистей стилей и разными цветами.

## Задачи для самостоятельной работы

1. Изобразить на экране отрезок, вращающийся в плоскости экрана вокруг:
  - а) своей середины;
  - б) своего конца.
2. Изобразить на экране два отрезка, вращающихся в плоскости экрана, каждый вокруг своего конца и со своей угловой скоростью.
3. Изобразить одновременное вращение двух стрелок – большой и малой, при котором одному полному обороту большой стрелки соответствует  $1/12$  оборота малой стрелки (как на циферблате часов). Стрелки можно для простоты заменить отрезками.
4. Изобразить на экране правильный треугольник, вращающийся в плоскости экрана вокруг своего центра.
5. Изобразить на экране разносторонний треугольник, вращающийся в плоскости экрана вокруг своего центра тяжести.
6. Изобразить на экране прямоугольник, вращающийся в плоскости экрана вокруг своего центра.
7. Изобразить на экране прямоугольник, вращающийся в плоскости экрана вокруг одной из своих вершин. Цвет фигуры должен изменяться при переходе от предыдущего положения к следующему.
8. Круглое кольцо вращается с постоянной угловой скоростью вокруг своего диаметра, расположенного параллельно горизонтальной оси экрана. Изобразить на экране процесс вращения. Считать, что в момент времени  $t$  кольцо выглядит для наблюдателя как эллипс, большая ось которого равна постоянной величине  $c$ , а малая равна  $c \cos \omega t$ , где  $\omega$  – угловая скорость вращения.
9. Изобразить равнобедренный треугольник, вращающийся с постоянной угловой скоростью вокруг своей высоты, расположенной параллельно вертикальной оси экрана.



10. Изобразите на экране движение шара по бильярду без луз. Используйте правило: угол отражения равен углу падения. Начальное направление движения должно задаваться.
11. Создайте мультфильм «Круги на воде», используя не менее семи концентрированных окружностей. Центры окружностей должны быть совмещены с центром окна, а радиусы изменяться, увеличиваясь с каждой следующей окружностью. Иллюзия движения должна создаваться последовательной сменой цветов всех окружностей, начиная с внутренней и кончая внешней. Процесс смены цветов должен происходить бесконечно.
12. Создайте мультфильм «Капли дождя». На экране в случайном месте должна появляться окружность, увеличиваться ее радиус и одновременно «выцветать». Процесс падающих капель должен завершаться по закрытию окна.
13. Составьте программу, которая выводит на экран текст в следующем виде:



- ☞ *Сохраните все программы в своей папке. Пригласите преподавателя и продемонстрируйте ему все выполненные вами задания. Будьте готовы ответить на вопросы преподавателя. После беседы с преподавателем выйдите из среды PascalABC.NET.*
- ☞ *Если вы обучаетесь в условиях использования дистанционных образовательных технологий, вышлите все файлы с подготовленными вами программами преподавателю.*

## Глава 16\*

# РАБОТА С ГРАФИЧЕСКИМИ ОБЪЕКТАМИ

### Задание 111

В электронном пособии ([Examples\Пособие\index.html](#)) изучите раздел «Основы объектно-ориентированного программирования». Ответьте на вопросы, приведенные в конце раздела.

В модуле GraphABC есть набор объектов, обеспечивающих возможность работы с рисунками формата bmp, jpg и gif. Познакомимся с технологией работы с подобными рисунками.

### Задание 112

Откройте файл `pic_01.pas`. Просмотрите текст программы, выясните, что она делает. Измените программу так, чтобы глобус двигался:

- а) по окружности;
- б) по периметру окна.

Фон окна должен представлять собой многоцветный рисунок.

### Задание 113\*

Напишите программу, эмулирующую классические часы со стрелками. Должен быть циферблат и три стрелки: часовая, минутная и секундная. У циферблата должен быть фон в виде картинки. Часы должны показывать реальное время!

#### *Подсказка*

Откройте файл `DigitalClock.pas` и выясните особенности работы с реальным временем (встроенным в компьютер таймером). Используйте эти возможности при разработке часов.

☞ *Как правило, если рисунок не выводится, это означает, что неверно указан путь к рисунку. Исправить это просто: указать полный путь или путь начиная с текущей папки.*

### Задание 114\*

Используя датчик случайных чисел, разработайте игру «Гонки». Суть игры: несколько объектов выходят на старт (автомобили, паровозы, лошади,

тараканы и т. п.). От игроков принимаются ставки, и затем участники гонок начинают движение. К финишу приходит первым кто-то один. Раздаются выигрыши и предлагается новый заезд.

☞ *Сохраните все программы в своей папке. Пригласите преподавателя и продемонстрируйте ему все выполненные вами задания. Будьте готовы ответить на вопросы преподавателя. После беседы с преподавателем выйдите из среды PascalABC.NET.*

☞ *Если вы обучаетесь в условиях использования дистанционных образовательных технологий, вышлите все файлы с подготовленными вами программами преподавателю.*

## Глава 17\*

# СОБЫТИЙНОЕ ПРОГРАММИРОВАНИЕ

До сих пор вы писали классические учебные программы, которые выполнялись последовательно: сначала первый оператор, затем второй и так до последнего. После выполнения последнего оператора программа завершала свою работу. Однако современные программы работают по-другому: после запуска они, конечно, тоже выполняют какие-то операторы, однако в основном их выполнение связано с ожиданием какого-либо действия пользователя, которое принято называть «событие». Обработка таких событий является ключевой задачей в написании современных программ.

Какие события может выполнить пользователь? Их количество не сильно велико: он может что-то сделать с помощью клавиатуры или с помощью мыши. Собственно, все.

### Задание 115

Откройте Справку, найдите статью «GraphABC: события» (раздел «Стандартные модули», подраздел «Модули GraphABC»). Познакомьтесь с описанием процедурных переменных.

Скопируйте на рабочие листы среды PascalABC.NET примеры событийных программ из Справки и выясните, как они работают. Добавьте в них событие, обеспечивающее завершение работы программы (реализуется с помощью процедуры CloseWindow).

### Задание 116

Выполните следующие действия:

- а) загрузите файл **event\_01.pas**;
- б) разберитесь в тексте программы;
- в) запустите программу на выполнение и выясните, что она делает;
- г) измените программу так, чтобы она:

завершала работу при щелчке *левой* клавиши мыши;



изменяла цвет звезды при щелчке *правой* клавиши мыши;  
вместо кружков рисовала квадратики;  
в качестве ведущей фигуры использовала 10-конечную звезду;  
ограничивала поле для рисования «следа» звезды в рамках прямоугольника с координатами диагонали (100, 100) и (300, 300).

### Задание 117


Составьте программу «Охота», в которой:

в произвольном месте окна появляется объект (кружок, звездочка, прямоугольник) на секунду (долю секунды);  
пользователь должен успеть щелкнуть мышкой по объекту;  
программа должна вести учет числа объектов и числа попаданий пользователем, причем числа должны быть выведены в заголовок окна;  
программа должна завершать свою работу по щелчку правой клавиши мыши.

### Задание 118



Составьте программу, дублирующую алгоритм, предложенный в **event\_01.pas**, но управление должно выполняться с помощью клавиш со стрелками () . Программа должна завершать работу по нажатию на клавишу .


### Задание 119


Составьте программу, при запуске которой какая-либо картинка движется по окружности по часовой стрелке, а при нажатии на клавишу  направление движения меняется на противоположное.

## Задачи для самостоятельной работы\*

1. Составьте программу – простейший графический редактор, с помощью которого можно рисовать не только линии, но и не менее 2 примитивов (скажем, окружность и прямоугольник) с помощью мыши. Программа должна напоминать простейший вариант программы Paint.

2. Составьте программу-раскраску (т. е. программу, при запуске которой на экране появляется черно-белый контурный рисунок и палитра цветов. Щелчком мыши в палитре выбирается цвет, а вторым щелчком по рисунку закрашивается контур, которому принадлежит выбранная точка.).
3. Составьте игру «Тир». Суть игры: на стенде случайным образом выскакивают фигурки, необходимо успеть щелкнуть мышью по фигурке. Идеальный вариант, если за курсором мыши движется прицел (например, такой: ).
4. Разработайте игру «Удав». Принцип работы: в окне появляется удав (точнее, его голова, например, в виде кружка. Но лучше нарисовать.). На экране в произвольном месте появляется «еда» (мелкие животные, цветные прямоугольники и т. п.). Удав движется сам, сначала, например, строго вверх. С помощью клавиш со стрелками () можем изменить направление движения. Если удав соприкасается с «едой», она исчезает, а игроку добавляются очки (в заголовке окна). Если удав касается окна, он «погибает». Игра заканчивается, скажем, после того, как удав съест 10 единиц «еды» или «погибнет».

 *Пригласите преподавателя и продемонстрируйте ему сделанные вами программы. Будьте готовы ответить на его вопросы.*

 *Если вы обучаетесь в условиях использования дистанционных образовательных технологий, вышлите все файлы с подготовленными вами программами преподавателю.*

## Глава 18

# ОСНОВНЫЕ ОПЕРАЦИИ СО СТРОКОВЫМИ ДАННЫМИ

### Задание 120

В электронном пособии (Examples\Пособие\index.html) изучите раздел «Основные операции со строками». Ответьте на вопросы, приведенные в конце раздела. Выполните тест по теме и продемонстрируйте оценку преподавателю.

## 1. Алгоритмы замены фрагментов строки

### Задание 121

Считайте с диска файл с именем `string_01.pas`. Он представляет собой программу поиска всех букв «а» и печать номеров позиций этих букв в строке символов. Запустите эту программу, введите строку, содержащую нужные символы, и убедитесь, что программа работает правильно.

Модифицируйте программу так, чтобы:

- а) подсчитывалось и печаталось число букв «а»;
- б) учитывались еще и заглавные буквы «А».

### Задание 122

Считайте с диска файл с именем **`string_02.pas`**. Он содержит программу замены всех букв «а» на «о».

Модифицируйте программу так, чтобы происходила замена буквы «а» на букву «б» и наоборот. Например, слово «баба» программа превращала бы в «абаб». Сделайте то же с учетом заглавных букв.

### Задание 123

В файле **`string_03.pas`** находится программа поиска в строке всех заданных фрагментов. Разберите, в чем ее отличие от предыдущих программ. Модифицируйте программу так, чтобы вместо фрагмента «мама» отыскивался фрагмент «ма» (в слове «мама» он должен обнаруживаться дважды).

### Задание 124

В файле **string\_04.pas** находится программа замены в строке одного слова другим. Разберите, в чем ее отличие от предыдущих программ.

Модифицируйте программу так, чтобы слово «мама» заменялось словом «бабушка». С какой проблемой вы столкнулись?

Напишите универсальную программу замены всех вхождений данного фрагмента на другой.

### Задание 125

Составьте программу замены одного слова строки другим.

Модифицируйте программу так, чтобы происходила замена только целого слова на слово (так чтобы, заменяя, например, «полковник» на «генерал», не получить нового звания «подгенерал»). Считайте, что слова разделяются пробелами.

При проверке работы программы обратите особое внимание на слова, стоящие в начале и в конце строки.

## 2. Алгоритмы распознавания слов и чисел в строке

### Задание 126

Напишите программу, подсчитывающую количество слов во введенной фразе:

- а) если между словами стоит один пробел;
- б) между словами может стоять произвольное число пробелов;
- в) между словами стоит один произвольный разделитель (пробел, «,», «.», «!», «?», «;», «-«);
- г) между словами стоит произвольное число разделителей.

### Задание 127

В файле **string\_05.pas** хранится программа, которая обеспечивает проверку того, является ли заданная строка целым числом. Разберитесь, как она работает. Проверьте, как работает программа, если на вход подается:

- а) целое число;
- б) строка, содержащая цифры и буквы (например, 123абв);



в) дробное число (например, 12.34);

г) посмотрите, что будет, если вместо константы '375' вставить '375a'.

Что изменится, если изменить описание переменной *i* (указать, например, «Real» вместо «Integer»)?

Используя эту идею, напишите программу, вычисляющую сумму всех целых чисел, находящихся во введенной строке.

### *Подсказка*

Для решения задачи достаточно вырезать все целые числа из строки, преобразовать их в числовые данные и добавить в сумму.

### Задание 128

Составьте простейший калькулятор, обеспечивающий ввод строки простейшим арифметическим выражением вида

$$a1 <оп> a2 <оп> a3 \dots <оп> aN,$$

где  $a1\dots aN$  – целые числа;  $<оп>$  – знак операции сложения (+) или вычитания (-) и вычисляющий результат описанных действий.

### Задание 129

В файле **string\_06.pas** хранится программа выделения и печати всех слов из введенной фразы, разделенных произвольным разделителем. Сравните ее с той, которую вы написали для решения этой задачи в предыдущей работе. Не содержится ли здесь новая для вас идея?

### Задание 130

Модифицируйте программу из файла **string\_06.pas** так, чтобы печатались только:

- а) слова, начинающиеся на букву «л»;
- б) слова, заканчивающиеся на букву «ь»;
- в) слова длиной не менее пяти букв;
- г) самое длинное слово.

## Задание 131

Подсчитайте, сколько пятерок потребуется, чтобы записать все числа от 1 до 999.

### Задачи для самостоятельной работы

1. Напишите программу, которая в заданном предложении находит самое короткое и самое длинное слово. Напечатайте исходный текст и найденные слова.
2. Напишите программу, которая в заданном предложении удаляет все символы пробела. Напечатайте исходный и преобразованный тексты, а также количество удаленных символов.
3. В качестве исходного текста задайте арифметическое выражение, записанное на Паскале. Напишите программу, которая проверяет, соблюдается ли равенство открывающих и закрывающих скобок. Выведите на печать исходный текст и количество открывающих и закрывающих скобок.
4. Напишите программы, которая в заданном предложении символы самого длинного слова заменяет символами «\*». Напечатайте преобразованный текст.
5. На вход программе подается набор символов, заканчивающихся точкой.

Напишите эффективную, в том числе и по используемой памяти, программу, которая сначала будет определять, есть ли в этом наборе символы, соответствующие десятичным цифрам. Если такие символы есть, то можно ли переставить их так, чтобы полученное число было симметричным (читалось одинаково как слева направо, так и справа налево). Ведущих нулей в числе быть не должно, исключение – число 0, запись которого содержит ровно один ноль.

Если требуемое число составить невозможно, то программа должна вывести на экран слово «NO». А если возможно, то в первой строке следует вывести «YES», а во второй – искомое симметричное число. Если таких чисел несколько, то программа должна выводить максимальное из них.

Например, пусть на вход подаются следующие символы:

Do not 911 to 09 do

В данном случае программа должна вывести:

YES

91019

6. Напишите программу, которая в заданном предложении перед каждым словом вставляет символ «?». Напечатайте преобразованный текст, а также длину полученного предложения.
  7. Напишите программу, которая в заданном предложении удаляет все запятые. Напечатайте преобразованный текст, а также число удаленных символов.
  8. В заданном предложении проверьте парность расстановки круглых скобок. Вложение скобок допускается.
  9. Напишите программу печати слов предложения в обратном порядке.
  10. Составьте программу, которую шифрует и дешифрует введенный текст. В качестве ключа шифра можно взять произвольную неповторяющуюся последовательность символов.
  11. Составьте программу, в которую вводится двузначное число и выводится его название (вводится «36» – выводится «тридцать шесть»).
  12. Составьте программу, проверяющую, является ли введенное предложение «перевертышем», т. е. читающимся одинаково слева направо и справа налево (без разделителей, разумеется). Примеры «перевертышей»: *Аргентина манит негра; а роза упала на лапу Азора.*
- ☞ *Сохраните все программы в вашей папке. Пригласите преподавателя и продемонстрируйте ему все сделанные вами программы. Будьте готовы ответить на вопросы преподавателя. После беседы с преподавателем выйдите из среды PascalABC.NET.*
- ☞ *Если вы обучаетесь в условиях использования дистанционных образовательных технологий, вышлите все файлы с подготовленными вами программами преподавателю.*

## Глава 19

# ИСПОЛЬЗОВАНИЕ ПРОЦЕДУР

### Задание 132

В электронном пособии ([Examples\Пособие\index.html](#)) изучите раздел «Процедуры». Ответьте на вопросы, приведенные в конце раздела. Выполните тест по теме и продемонстрируйте оценку преподавателю.

Рассмотрим использование процедур на примерах.

### Задание 133

1. Запустите среду **PascalABC.NET**.
2. Загрузите файл **Proc5.pas**.
3. Проанализируйте текст программы. Это сделать будет совсем не трудно, т. к. вы уже знакомы с алгоритмом поиска максимального из двух чисел.
4. Найдите описание процедуры. Ответьте на следующие вопросы:
  - а) какое у процедуры имя;
  - б) сколько параметров у процедуры;
  - в) какие из них входные, выходные;
  - г) найдите конец процедуры. Чем он отличается от конца программы?
5. Найдите тело головной программы. Сколько раз вызывается процедура? Посмотрите, как описаны фактические параметры. Совпадает ли это с описанием формальных параметров? Если найдете несоответствие в описании параметров, то обратитесь к преподавателю.
6. Запустите программу на выполнение. Проверьте, правильно ли она работает.
7. Модифицируйте программу так, чтобы она работала и для шести чисел.
8. Заставьте программу находить минимальное из шести заданных чисел.

### Задание 134

Загрузите файл **Proc6.pas**. В файле приведена программа, в которой идет отладка процедур рисования дирижабля и мишени. Используя данный пример как основу, разработайте процедуру, которая рисует:

- |               |            |
|---------------|------------|
| а) дерево;    | г) машину; |
| б) человечка; | д) солнце; |
| в) цветок;    | е) забор.  |

(Выберите, что вам больше нравится, и подумайте, как можно будет задавать параметры этих объектов: размер, цвет и т. п.)

Текст доработанной программы с добавленной процедурой запишите с новым именем в свою папку.

### Задание 135

1. Загрузите файл **Proc7.pas**.
2. Перед вами программа **TOWN**: она строит город из домов. Проанализируйте структуру программы. Она содержит процедуру **HOUSE**, которая умеет строить дом по заданным параметрам. Посмотрите, какие параметры имеет процедура.
3. Найдите вызовы процедуры. Сколько, на ваш взгляд, будет построено домов?
4. Запустите программу на выполнение и проверьте свой ответ.
5. Придумайте свой город (свою комбинацию домов) и, исправив в головной программе вызовы процедуры, получите свою картинку. Покажите ее преподавателю.

### Задание 136

1. В одном из предыдущих заданий вы составили свою процедуру с изображением некоторых объектов (деревья, человечки, цветы, солнце). Оформите ее в виде процедуры и добавьте в программу **TOWN**.
2. Если у вашего соседа по компьютерному классу своя процедура для рисования объектов, то возьмите ее у него. Обратите внимание: если

оформлять алгоритмы в виде процедур, то одну программу могут разрабатывать сразу несколько человек. Это очень удобно!

### **Задание 137**

Большое число практических задач требует каким-то образом усреднять данные. Это и обработка результатов научных экспериментов, и задачи статистики, и др.

Самое простое – взять среднее арифметическое всей последовательности чисел. Другой способ – взять среднее арифметическое между наибольшим и наименьшим элементами последовательности. Еще один способ, более сложный, применяется, в частности, при судействе соревнований по фигурному катанию. Состоит он в том, чтобы отбросить крайние (наибольшее и наименьшее) числа, а от остальных взять среднее арифметическое.

Запрограммируйте все эти способы для вводимой последовательности чисел с помощью отдельных процедур. Сравните результаты.

### **Задачи для самостоятельной работы**

1. Часто бывает важно не только определить среднее значение последовательности, но и охарактеризовать величину разброса. Одной из таких характеристик может служить, например, разность между наибольшим и наименьшим значениями. С помощью процедуры запрограммируйте вычисление этой величины для вводимой последовательности чисел.
2. Составьте процедуру, с помощью которой можно будет определять расстояние между двумя точками в трехмерном пространстве.
3. Разработайте процедуру, с помощью которой определяется, является ли введенное число простым.
4. Составьте процедуру, которая вместо даты, заданной в числовом виде (25.06), выводит ее текстовое название (двадцать пятое июня).
5. Разработайте процедуру, которая преобразует дату из классического вида (15 мая 2011 года) в числовой (15.05.2011).

6. Составьте процедуру, которая приводит строку, состоящую из русских символов, к одному регистру (например, прописные буквы преобразует в строчные).  
Составьте процедуру, которая определяет в заданном предложении все слова-перевертыши (*казак, шалаш*).
7. Составьте процедуру выравнивания строки, которая преобразует ее к строке указанной длины путем добавления пробелов между словами.
8. Составьте процедуру, с помощью которой из заданного текста будут удалены все английские символы.
9. Составьте процедуру, являющуюся синтаксическим анализатором идентификатора. Процедура должна получать строку и проверять, является ли полученная строка идентификатором (т. е. последовательностью символов, состоящую из английских букв, цифр и (или) знака подчеркивания, начинающуюся с буквы).
10. Разработайте процедуру, которая рисует в заданном месте произвольный закрашенный треугольник.
11. Разработайте процедуру, которая рисует параллелепипед по заданным параметрам.
12. Разработайте процедуру **LineRel(xs,ys)**, которая рисует отрезок, соединяющий текущую точку и точку, отстоящую от текущей на заданное смещение **(xs, ys)** (т. е. координаты второй точки равны **(x+xs, y+ys)**).

☞ *Пригласите преподавателя и продемонстрируйте ему сделанные вами программы и процедуры. Будьте готовы ответить на вопросы преподавателя. После беседы с преподавателем выйдите из среды PascalABC.NET.*

☞ *Если вы обучаетесь в условиях использования дистанционных образовательных технологий, вышлите все файлы с подготовленными вами программами преподавателю.*

## Глава 20

### РАБОТА С МАССИВАМИ: ПЕРВЫЕ ШАГИ

#### Задание 138

В электронном пособии (Examples\Пособие\index.html) изучите раздел «Массивы». Ответьте на вопросы, приведенные в конце раздела. Выполните тест по теме и продемонстрируйте оценку преподавателю.

#### Задание 139

Запустите среду PascalABC.NET.

Составить программу:

- а) в которой всем элементам массива присваивается заданное целое значение (количество элементов массива не более 50);
- б) всем элементам массива присваивается вводимый текст;
- в) каждому элементу массива с нечетным индексом присваивается его номер, а элементу с нечетным индексом – его номер с противоположным знаком;
- г) элементам массива, стоящим на четных местах, присваивается символ «@», на нечетных местах – «#».
- д) каждому элементу, стоящему на первом, четвертом, седьмом и т. д. местах, присваиваться число 0, остальным – число 3, и все элементы массива выводятся на экран дисплея.

#### Задание 140

Загрузите в новое окно программу из файла **array\_1.pas**. Эта программа позволяет ввести последовательность из 15 чисел и вывести их в обратном порядке. Разберитесь в том, как работает эта программа. Выполните задание, написанное в конце программы (после END.)

#### Задание 141

Модифицируйте программу из файла **array\_1.pas** так, чтобы сначала печатались в строчку все положительные числа, а в следующей строке – все отрицательные.



#### Задание 142

В массиве хранится информация о количестве осадков за каждый день ноября. Составьте программу, с помощью которой определяется общее количество осадков за этот месяц.

#### Задание 143

В массиве хранится информация о количестве студентов в каждой группе. Составьте программу, с помощью которой определяется среднее число студентов в одной группе.

#### Задание 144

В массиве хранится информация о результатах квалификационных выступлений фигуристов. На соревнования допускаются только те, кто получил результаты выше среднего. Составьте программу, которая определяет число спортсменов, прошедших квалификацию.

#### Задание 145

В массиве хранится информация о результатах соревнований по прыжкам в длину. Составьте программу, с помощью которой определяется лучший и худший из результатов.

#### Задание 146

В массиве содержатся оценки ученика по математике. Сосчитайте общее количество пятерок, четверок, троек и двоек.

#### Задание 147

Представьте себе, что вы смешиваете несколько компонентов в одну смесь (может быть, создаете новые духи, может быть, особо прочный и легкий сплав металлов и т. п.). По окончании работы вы знаете, сколько чего входит в смесь, т. е. массу каждого компонента. Чтобы в дальнейшем изготавливать такую же смесь в промышленных масштабах, вам нужен «рецепт», т. е. процентное содержание каждого компонента в смеси.

Такая программа составлена и размещена в файле **array\_2.pas**. Загрузите этот файл. Модифицируйте программу, размещенную в этом файле, так, чтобы она запрашивала не только массы, но и названия компонентов и печатала их

в рецепте. Имейте в виду, что элементами массива могут быть переменные любого типа, в том числе и строковые. Для этой программы вам потребуется два массива – один, как и прежде, числовой (**Real**) для хранения масс, а другой – строковый (**String[...]**) для хранения названий.

#### Задание 148

Загрузите файл **array\_char\_01.pas**. Закончите данную программу, выполнив задание, предложенное в начале программы. Считайте, что в тексте присутствуют либо маленькие (строчные) английские буквы, либо «не буквы».

#### Задание 149

Программа из предыдущего задания работает без учета больших букв. Измените программу так, чтобы она учитывала все (большие и маленькие) буквы.

#### *Подсказка*

Самый простой путь решения – свести все буквы к одному виду – большим или маленьким. Для этих целей существуют специальные функции **UpCase** и **LowerCase**. Посмотри ее описание в Справке (Помощь → Справочник по языку → Системные процедуры, функции, типы и константы → Процедуры и функции для работы с символами и строками).

#### Задачи для самостоятельной работы

1. В массиве рост юношей задан отрицательным числом, рост девушек – положительным. Определите отдельно средний рост юношей и девушек.
2. В массиве хранится информация о стоимости книг. Составьте программу, которая выводит на экран номера книг, стоящих ниже средней стоимости всех этих книг.
3. В массиве хранится число жителей в каждом доме улицы. Индекс элемента массива соответствует номеру дома. Известно, что на одной стороне улицы дома с четными номерами, а с другой – с нечетными. Составьте программу, которая определяет, на какой стороне улицы больше жителей: на четной или нечетной.
4. Определите, имеются ли в заданном массиве одинаковые элементы.

5. Определите, является ли заданный массив упорядоченным по возрастанию. При отрицательном ответе выведите номер элемента, нарушающего эту упорядоченность.
6. В массиве хранятся года рождения некоторого числа людей. Составьте программу, которая определит разницу в возрасте между самым старшим и самым младшим из этого списка. Считать, что возраст можно определить как разность между текущим годом и годом рождения.
7. В массиве хранится информация о стоимости различных марок автомобилей. Определите два самых дорогих автомобиля из этого списка. Попытайтесь решить эту задачу, используя один проход по массиву.
8. Удалите из введенного файла отрицательные элементы. В конец массива впишите нули.
9. Известна стоимость автомобилей и их мощность (л. с.). Определить общую стоимость автомобилей, имеющих мощность больше 100 л. с.
10. Составьте программу, которая вычисляет частоту использования русских букв во введенном тексте.  
Протестируйте разработанную программу. Для этого запустите ее:
  - а) сразу нажмите . Программа должна сообщить, что текст отсутствует;
  - б) введите текст «vvv». Программа должна сообщить, что в тексте русских букв нет;
  - в) введите «ююююю». Программа должна выдать только одну строчку: букв «ю» в тексте 100 %;
  - г) введите «мама». Программа должна выдать две строчки: букв «а» в тексте 50 % ; букв «м» в тексте 50 % ;
  - д) введите «123 АбвГ123деЖзий». Программа должна выдать информацию о том, что букв а, б, ..., й в тексте по 10 %.
11. Для характеристики разброса элементов последовательности в статистике используется величина, называемая дисперсией.

*Дисперсия* – это среднее арифметическое квадратов разностей между элементами последовательности и ее средним значением, вычисляется по формуле

$$D = \frac{(a_1 - M)^2 + \dots + (a_n - M)^2}{n},$$

где  $a_i$  – элементы последовательности, а  $M$  – среднее значение этих элементов.

Напишите программу для вычисления дисперсии (и обязательно сохраните – она нам еще понадобится).

12. Для иллюстрации дискретных динамических процессов довольно часто используются столбиковые диаграммы (гистограммы) (рис. 4).

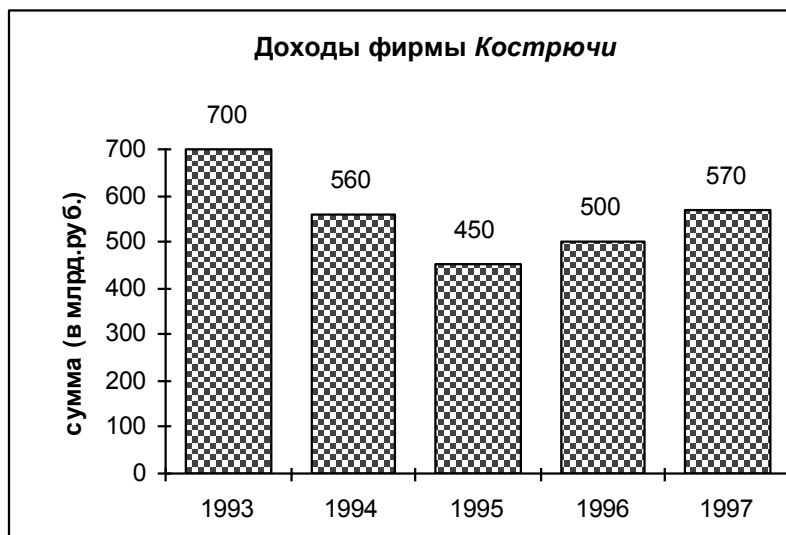


Рис. 4. Пример столбиковой диаграммы (гистограммы)

Составьте программу, которая по введенным данным строила бы такую гистограмму на экране. Сделайте 2 варианта программы: из двумерных столбиков (рис. 4) и трехмерных параллелепипедов (рис. 5).

#### *Уточнение*

Программа будет универсальной только в том случае, если она будет работать для произвольного (например, от 1 до 100) набора чисел и для любых значений этих чисел. Чтобы столбики были пропорциональны и помещались на экране, требуется пропорционально увеличивать размеры столбиков, если значения чисел малы, и уменьшать, если они слишком велики (этот процесс назовем масштабированием).

Цвета каждого из столбиков и подписи под ними также следует вводить.

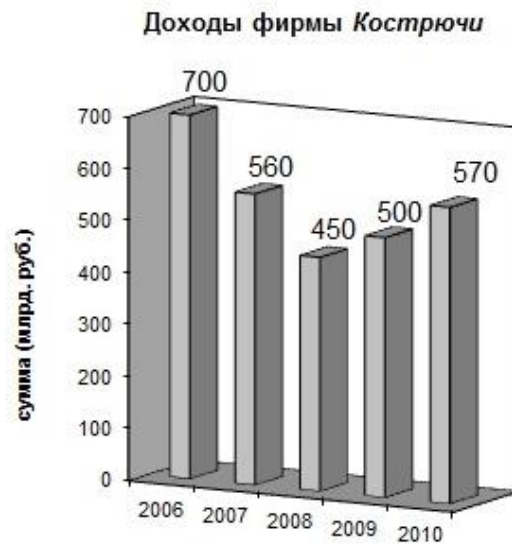


Рис. 5. Пример трехмерной гистограммы

13. Напишите программу, которая строит круговую диаграмму, интерпретирующую последовательность заданных чисел (рис. 6).

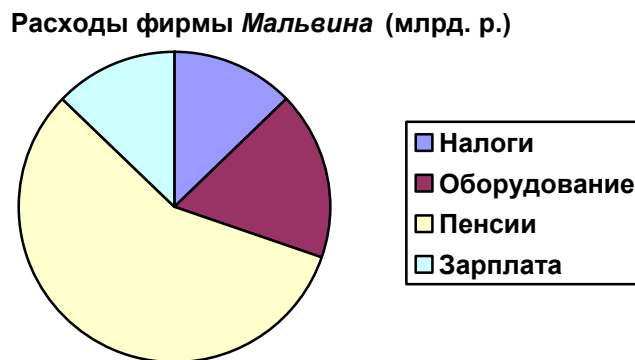


Рис. 6. Пример круговой диаграммы

#### *Подсказка*

Предположим, заданы числа 3, 5 и 2. Сумма этих чисел:  $3 + 5 + 2 = 10$ . Следовательно, число 3 составляет 30 % от суммы, 5 – 50 % и 2 – 20 %; 30 % окружности – это  $360 \cdot 0,3 = 108^\circ$ , 50 % –  $180^\circ$  и 20 % –  $72^\circ$ . В этом случае вам следует построить окружность из 3 разноцветных секторов, значение которых соответствует  $108^\circ$ ,  $180^\circ$  и  $72^\circ$ . Это и будет круговая диаграмма, соответствующая заданной последовательности чисел.

Решите задачу в общем случае.



14. Составьте программу, которая хранит информацию о вкладчиках банка. Программа должна по запросу выдавать:
- общую сумму всех вкладов зарегистрированных вкладчиков;

- б) список вкладчиков, размеры их вкладов и сумму денег, которые они будут получать в виде процентов, поскольку за хранение денег в банке вкладчикам полагается процент. В зависимости от вида вклада этот процент колеблется, например, от 1 до 10 %;
- в) информацию о конкретном вкладчике.

15. Напишите программу, которая вводит фамилию и возраст посетителей поликлиники, а по окончании ввода печатает отдельные списки больных до 18 лет, от 18 до 50 лет и старше 50 лет.

#### *Подсказка*

По-видимому, в этой задаче вам понадобится один массив для хранения возраста, а еще один – для хранения фамилий.

-  *Пригласите преподавателя и продемонстрируйте ему сделанные вами программы. Будьте готовы ответить на вопросы преподавателя.*
-  *Если вы обучаетесь в условиях использования дистанционных образовательных технологий, вышлите все файлы с подготовленными вами программами преподавателю.*

## Глава 21\*

# СОРТИРОВКА МАССИВОВ

### Задание 150

В электронном пособии ([Examples\Пособие\index.html](#)) изучите раздел «Сортировка массивов». Ответьте на вопросы, приведенные в конце раздела.

### Задание 151

Имеется список спортсменов, участвующих в соревнованиях по прыжкам в длину, и их результаты. Составить программу, которая позволит вывести список участников на экран дисплея в порядке занятых ими мест.

### Задание 152

Введите и отладьте программу сортировки числового массива по возрастанию методом «всплывающего пузырька». Для того чтобы увидеть результаты работы программы, включите в нее ввод данных и печать результатов.

### Задание 153

Измените программу так, чтобы массив сортировался в порядке убывания.

### Задание 154

Еще одна модификация: замените числовой массив строковым и отсортируйте его в алфавитном порядке.

### Задание 155\*

Напишите алгоритм слияния двух массивов, отсортированных в порядке возрастания, в один массив, также отсортированный в порядке возрастания. Самый простой способ решения этой задачи состоит в том, чтобы попросту «склеить» эти два массива, а затем отсортировать получившийся массив, например, методом пузырька. Подсчитайте количество сравнений, которые придется выполнить.

Другой путь – сразу помещать элементы в новый массив в нужном порядке, пользуясь тем, что исходные массивы уже упорядочены. Реализуйте эту идею. Подсчитайте количество сравнений и сопоставьте с предыдущим результатом. Понятно, что этот путь предпочтительнее.

### Задание 156\*

Составить программу обслуживания конькобежных соревнований.

Программа должна:


- 1) обеспечивать ввод фамилии спортсмена и его результат;
- 2) выводить после каждого ввода информации о спортсмене полного списка спортсменов (фамилии и результат) в порядке занятых мест.


### Задачи для самостоятельной работы

1. Задан массив оценок ученика по предмету. Составьте программу, которая позволяет ввести эти оценки в массив и отсортировать их так, чтобы в начале списка были пятерки, далее – четверки, затем – тройки и двойки.
2. Задан массив, содержащий оценки фигуристки. Составить программу, которая удаляет из массива максимальный и минимальный элементы и выводит оставшиеся результаты подряд в порядке убывания.
3. Дан массив, содержащий список должностей всех сотрудников фирмы. Разумеется, в этом списке некоторые должности могут повторяться. Нужно составить штатное расписание, т. е. сделать из имеющегося списка два: в первом – название всех должностей (каждая из должностей в списке может встречаться только один раз), во втором – количество занятых ставок каждой из этих должностей. Массив должностей должен быть отсортирован по алфавиту.
4. В массиве содержится информация о росте студентов группы. Массив упорядочен по убыванию. В группу перевели новенького. Составьте программу, которая в имеющийся массив вставит рост новичка так, чтобы массив остался также упорядоченным.
5. *Разработать процедуру, обеспечивающую реализацию алгоритма сортировки вставками.* Дана последовательность чисел  $a_1, a_2, \dots, a_n$ . Требуется переставить числа в порядке возрастания. Делается это следующим образом. Пусть  $a_1, a_2, \dots, a_i$  – упорядоченная последовательность, т. е.  $a_1 \leq a_2 \leq \dots \leq a_i$ . Берется следующее число  $(a_i + 1)$ , и вставляется в последовательность так, чтобы новая последовательность была также возрастающей. Процесс производится до тех пор, пока все элементы от  $(i + 1)$  до  $n$  не будут перебраны.



6. *Разработать процедуру, обеспечивающую реализацию алгоритма сортировки Шелла. Дан массив  $n$  действительных чисел. Требуется упорядочить его по возрастанию. Делается это следующим образом: сравниваются два соседних элемента  $a_i$  и  $(a_i + 1)$ . Если  $a_i \leq a_i + 1$ , то продвигаются на один элемент вперед. Если  $a_i > a_i + 1$ , то производится перестановка и сдвигаются на один элемент назад. Составить алгоритм этой сортировки.*

 *Пригласите преподавателя и продемонстрируйте ему сделанные вами программы. Будьте готовы ответить на вопросы преподавателя.*

 *Если вы обучаетесь в условиях использования дистанционных образовательных технологий, вышлите все файлы с подготовленными программами преподавателю.*

## Глава 22\*

# МНОГОМЕРНЫЕ МАССИВЫ

### Задание 157

В электронном пособии ([Examples\Пособие\index.html](#)) изучите раздел «Многомерные массивы». Ответьте на вопросы, приведенные в конце раздела. Выполните тест по теме и продемонстрируйте оценку преподавателю.

### Задание 158

Написать программу, которая вводит по строкам с клавиатуры двумерный массив и вычисляет сумму его элементов по строкам.

### Задание 159

Написать программу, которая вводит по строкам с клавиатуры двумерный массив и вычисляет сумму его элементов по столбцам.

### Задание 160\*

В программу обслуживания шахматного турнира целесообразно включить процедуры ввода данных после каждой встречи и печати турнирной таблицы. Напишите эти процедуры, используя заготовки, рассмотренные в теории. Напишите и отладьте головную программу, вызывающую эти процедуры.

### Задание 161\*

Обеспечьте выдачу всей турнирной таблицы и выдачу списка участников турнира, упорядоченную по количеству набранных ими очков.

### Задание 162\*

Предусмотрите также возможность упорядочить таблицу так, чтобы фамилии участников стояли в алфавитном порядке.

### Задание 163

Имеется таблица, включающая фамилии студентов и оценки за 5 экзаменов.

Составить программу:

- а) вывода фамилий отличников;
- б) «        «        студентов, не имеющих оценки «неуд»;
- в) «        «        «        имеющих только оценки «уд»;

- г) вывода фамилий студентов, обучающихся только на «хорошо» и «отлично»;
- д) «                »                «                » имеющих не более одной «уд»;
- е) вывода фамилий учащихся, имеющих двойки;
- ж) определения номера предмета, по которому учащиеся учатся хуже всех;
- и) то же, лучше всех.

### Задание 164

Дана следующая таблица, содержащая данные о планетах Солнечной системы:

Планета	Расстояние до Солнца	Относит. объем	Относит. масса
Меркурий	57,9	0,06	0,05
Венера	108,2	0,92	0,81
Земля	149,6	1,00	1,00
Марс	227,9	0,15	0,11
Юпитер	978,3	1345,00	318,40
Сатурн	1429,3	767,00	95,20

Напишите программу, которая определяет параметры:

- наиболее и наименее удаленные от Солнца планеты, а данные таблицы сортирует по возрастанию масс планет;
- средний объем планет Солнечной системы и находит среди них ту, объем которой наименее отличается от среднего. Сортирует данные таблицы по возрастанию расстояний до Солнца;
- самую тяжелую и самую удаленную от Солнца планеты. Сортирует данные таблицы по возрастанию объемов планет.

### Задачи для самостоятельной работы<sup>3</sup>

- Дана целочисленная таблица размером  $m \times n$ , в которой имеется два одинаковых элемента. Найти индексы этих элементов.

<sup>3</sup> Использованы задачи из электронных сборников: Программирование / С. А. Абрамов [и др.]. URL: [http://math.ivanovo.ac.ru/science/textbooks/Schanicina2006/I\\_Pascal/abramov\\_zadachi.pdf](http://math.ivanovo.ac.ru/science/textbooks/Schanicina2006/I_Pascal/abramov_zadachi.pdf) ; Паскаль – задачи и решения. URL: <http://programm.ws/page.php?id=494> (дата обращения 29.05.2014).

2. Написать программу, которая проверяет, является ли введенная с клавиатуры квадратная матрица магическим квадратом. *Магическим квадратом* называется матрица, сумма элементов которой в каждой строке, в каждом столбце и по каждой диагонали одинакова (см. приведенный ниже рисунок).

2	9	4
7	5	3
6	1	8

13	8	12	1
2	11	7	14
3	10	6	15
16	5	9	4

3. *Латинским квадратом* порядка  $n$  называется квадратная таблица размером  $n \times n$ , каждая строка и каждый столбец которой содержит числа  $1, 2, \dots, n$ . Составить программу, которая обеспечивает ввод целочисленной квадратной таблицы и проверку того, является ли она латинским квадратом.
4. Написать программу игры «Угадай число». Правила игры следующие. Игроют двое. Один задумывает число, второй угадывает. На каждом шаге угадывающий делает предположение, а задумавший число говорит, сколько цифр числа угаданы и сколько из этих цифр занимают правильные позиции в числе. Например, если задумано число 725 и выдвинуто предположение, что задумано число 523, то считаются угаданными две цифры (5 и 2) и одна из них (2) занимает верную позицию. Ниже представлен рекомендуемый вид экрана во время работы программы.

Компьютер задумал трехзначное число. Вы должны его отгадать. После очередного числа вам будет сообщено, сколько цифр угадано и сколько из них находится на своих местах. После ввода числа нажимайте *Enter*. Для завершения игры нажмите *Enter*.

Ваш вариант -> 123 Угадано: 0.На своих местах:0

Ваш вариант -> 456 Угадано: 1.На своих местах:0

Ваш вариант -> 654 Угадано: 2.На своих местах:2

Ваш вариант -> 657 Угадано: 2.На своих местах:2

Ваш вариант -> 658 Угадано: 3.На своих местах:3

\*\*\* ВЫ УГАДАЛИ ! \*\*\*

Нажмите *Enter* для завершения.

5. Написать программу-телеграф, которая принимает от пользователя сообщение и выводит его на экран в виде последовательности точек и тире. Вывод точек и тире можно сопроводить звуковым сигналом соответствующей длительности. Азбука Морзе для букв русского алфавита приведена ниже.

а	.-	б	-...	в	.---	г	--.
д	-..	е		ж	...-	з	--..
и	..	и	.---	к	-.-	л	.-..
м	--	н	-.	о	---	п	.-.
р	.-.	с	...	т	-	у	..-
ф	...-	х	....	ц	-.-.	ч	---.
ш	---	щ	--.-	ъ	...-	ы	-.—
ь	...-	э	...-	ю	..--	я	.-.-

☞ *Пригласите преподавателя и продемонстрируйте ему сделанные вами программы. Будьте готовы ответить на вопросы преподавателя.*

☞ *Если вы обучаетесь в условиях использования дистанционных образовательных технологий, вышлите все файлы с подготовленными вами программами преподавателю.*

## Глава 23

### ПОДПРОГРАММЫ-ФУНКЦИИ

Подпрограммы-функции предназначены преимущественно для вычисления какого-либо значения (числовое или текстовое). У них три основных отличия от процедур.

Первое отличие – в записи заголовка. Заголовок функции имеет следующую структуру:

**function** <имя>[(<список параметров>)]:<тип результата>;

Функция может и не иметь параметров.

Второе отличие заключается в том, что в теле функции хотя бы раз должно быть присвоено значение имени функции. Как правило, это значение и есть результат работы функции.

Третье отличие заключается в том, что результат функции должен быть использован либо в арифметическом, либо в логическом выражении.

Функции так же, как и процедуры, описываются в блоке описания данных (как правило, после описания переменных). Общая структура функции:

**function** <имя>[(<список параметров>)]:<тип результата>;

<описание данных>

**Begin**

<тело функции>

**end;**

Рассмотрим пример вычисления факториала числа  $N$ . Известно, что  $N! = N \cdot (N - 1) \cdot (N - 2) \cdot (N - 3) \cdot \dots \cdot 2 \cdot 1$ .

Подпрограмма-функция для вычисления факториала может иметь следующий вид:

```
Function Factorial(N:Integer) : Integer;  
var F,I : Integer;  
begin  
  F:=N;
```

```

for i:=N-1 downto 2 do F := F*i;
Factorial:=F
end;

```

Для вызова функции из основной программы или другой подпрограммы следует в выражении, где необходимо использовать значение функции, указать имя функции со списком фактических параметров, которые должны совпадать по количеству и типам с формальными параметрами. Так вызов вышеописанной функции Factorial может иметь следующий вид:

```
Part:=Factorial(20)/b;
```

Вызывается функция Factorial с фактическим параметром 20 (параметр-значение)

```
Part:=Factorial(M)+c/2;
```

Вызывается функция Factorial с фактическим параметром M (параметр-переменная)

```
Part:=Factorial(M+L);
```

В качестве параметра передается выражение, значение которого должно иметь тот же тип, что и описанный в функции формальный параметр.

### Задание 165

Запустите среду PascalABC.NET.

Составьте программу вычисления величины  $S$  по формуле  $S = F(b) - F(a)$ , где

$$F(x) = \begin{cases} \cos x & \text{при } x \leq 0, \\ \ln(x+1), & \text{если } x > 0. \end{cases}$$

Вычисление функции  $F(x)$  оформите в виде подпрограммы-функции. Подберите тесты для проверки правильности работы программы. Если убедитесь, что ваша программа работает корректно, покажите ее преподавателю.

### Задание 166

Вычисление значения функции  $y = x^n$  может быть оформлено в виде рекурсивной функции:

$$x^n = \begin{cases} 1, & \text{если } n = 0, \\ xx^{n-1}, & \text{если } n > 0. \end{cases}$$

Выполните следующие действия:

- а) загрузите файл **lab23\_1.pas** и посмотрите, как это записано в виде рекурсивной подпрограммы-функции;
- б) используя эту функцию, напишите программу, которая для заданных значений  $N$  и  $M$  вычисляет и выводит на экран значение функции

$$z = x^n + y^n;$$

- в) попробуйте переделать подпрограмму-функцию таким образом, чтобы в ней не было рекурсии. Можно использовать простую итерацию:

$$x^n = x \cdot x \cdot x \cdot \dots \cdot x \quad (n - \text{раз});$$

- г) запустите на выполнение программу с использованием своей новой процедуры. Оцените время работы программы и память.

### Задание 167

Наверное, все вы умеете находить наибольшее кратное двух чисел. Попробуйте составить подпрограмму-функцию, которая будет находить наибольший общий делитель для двух положительных целых чисел.

### Задание 168

Сравните свою подпрограмму-функцию с той, которая находится в файле **lab23\_2.pas**. Составьте схему работы функции для чисел 9 и 12. Сколько раз будет вызываться функция? С какими аргументами? Попробуйте поработать с отладчиком и проверить свои рассуждения.

### Задание 169


Составьте программу, которая позволяла бы определять наибольший общий делитель для трех заданных чисел.


## Задачи для самостоятельной работы

1. Написать функцию, которая вычисляет объем цилиндра. Параметрами функции должны быть радиус и высота цилиндра.



2. Написать функцию, которая сравнивает два целых числа и возвращает результат сравнения в виде одного из знаков:  $>$ ,  $<$  или  $=$ .
3. Написать функцию, которая вычисляет значение  $ax$ . Числа  $a$  и  $x$  могут быть любыми дробными положительными числами.
4. Написать функцию `Dohod`, которая вычисляет доход по вкладу. Исходными данными для функции являются: величина вклада, процентная ставка (годовых) и срок вклада (количество дней).
5. Написать функцию `Glasn`, которая возвращает значение `True`, если символ, полученный функцией в качестве аргумента, является гласной буквой русского алфавита.
6. Написать функцию `LTrim`, которая удаляет начальные пробелы из строки, полученной в качестве аргумента.
7. Написать функцию `Sogi`, которая возвращает `True`, если символ, полученный функцией в качестве аргумента, является согласной буквой русского алфавита.
8. Написать функцию, обеспечивающую ввод с клавиатуры целого положительного двухзначного числа. При нажатии пользователем какой-либо клавиши соответствующий ей символ должен появляться на экране только в том случае, если этот символ является цифрой. Функция должна позволять пользователю редактировать введенное число при помощи клавиши. Если пользователь ввел хотя бы одну цифру, то при нажатии клавиши функция должна завершать работу и возвращать введенное пользователем число вызвавшей ее процедуре.

 *Пригласите преподавателя и продемонстрируйте ему сделанные вами программы. Будьте готовы ответить на вопросы преподавателя.*

 *Если вы обучаетесь в условиях использования дистанционных образовательных технологий, вышлите все файлы с подготовленными вами программами преподавателю.*

## Глава 24

# РАБОТА С ТЕКСТОВЫМИ ФАЙЛАМИ:

## ЧТЕНИЕ И ЗАПИСЬ

До сих пор были рассмотрены способы работы с информацией, которая либо была заложена в программу, либо вводилась с клавиатуры компьютера. В данной лабораторной работе предлагается познакомиться с обработкой информации, размещенной в файлах на дисках.

В языке Паскаль предусмотрена работа с тремя типами файлов:

- 1) с текстовыми файлами;
- 2) типизированными;
- 3) нетипизированными.

Рассмотрим работу с двумя первыми видами<sup>4</sup>.

### 1. Понятие о текстовом файле

Паскаль позволяет работать с *текстовыми файлами*. Текстовые файлы содержат текстовую информацию в виде отдельных порций, называемых *строками*. Строки состоят из отдельных символов и могут иметь различную длину. Для отделения строк друг от друга используется специальный символ *конец строки* EOLN (End Of LiNe), который записывается в файл автоматически после нажатия клавиши Enter или при использовании команды **WriteLn** (при программном заполнении файла). В конце файла записывается символ *конец файла* (End Of File). Учитывая, что информация на диске располагается последовательно, байт за байтом, общую структуру текстовых файлов можно изобразить следующим образом:

Строка1	[EOLN]	Строка2	[EOLN]	Строка3...	[EOF ]
---------	--------	---------	--------	------------	-----------

---

<sup>4</sup> Освоив работу с текстовыми и типизированными файлами, вы без труда научитесь работать с нетипизированными файлами с помощью любого справочника.

Учитывая, что «строка» – понятие неоднозначное (в ней может быть как один символ, так и несколько тысяч), работа с текстовыми файлами осуществляется методом *последовательного доступа*: чтобы добраться до нужной порции информации (строки), необходимо прочитать все предыдущие строки от начала файла до нужного места.

## 2. Принцип работы с текстовыми файлами

При работе с текстовыми файлами необходимо выполнять следующую последовательность действий.

1. В блоке описания переменных VAR с помощью ключевого слова **text** описывается специальная *файловая переменная-указатель*. Например:

**var** UTF : **text**;

2. В программе файловая переменная связывается с конкретным файлом на диске<sup>5</sup>. Для этого используется специальная процедура **Assign** следующей структуры:

**Assign**(<имя файловой переменной>, <имя файла>);

или

**AssignFile**(<имя файловой переменной>, <имя файла>);

3. Для открытия файла используется одна из следующих стандартных процедур:

**Reset**(<имя>) – открыть текстовый файл только для чтения;

**Rewrite**(<имя>) – «            «            « только для записи;

**Append**(<имя>) – «            «            « только для дописывания текста в конец файла.

☞ *Полный перечень стандартных процедур и функций работы с файлом смотрите в Справке среды PascalABC.NET (Помощь→ Справочник*

---

<sup>5</sup> С точки зрения технологии файловая переменная – это некий буфер в оперативной памяти, в который считывается часть файла и с которым программа и работает. Запись также производится в буфер, и, когда буфер заполняется, его содержимое «сбрасывается» в файл на диске.

по языку→ Стандартные процедуры и функции→ Процедуры и функции для работы с файлами).

4. В случае, если файл открыт для чтения, используются стандартные процедуры **Read** и **ReadLn**. Их использование отличается от традиционного тем, что первым элементом в списке ввода стоит имя файловой переменной. Например:

```
...  
ReadLn(OUTF,A,B,C);  
...
```

☞ Процедура *Read* позволяет читать символы один за другим. Процедура *ReadLn* читает строку, а затем переводит курсор в начало следующей строки (даже если все не прочитано).

5. В случае, если файл открыт для записи, используются стандартные процедуры **Write** и **WriteLn**. В них также требуется указать первым в списке вывода имя файловой переменной. Например:

```
...  
WriteLn(OUTF,'А это будет записано в файл ...');  
...
```

☞ Процедура *Write* обеспечивает запись данных в текущую строку. Процедура *WriteLn* записывает строку и в конце ее ставит символ конца строки (EOLN). В этом случае следующие данные будут записываться в новой строке текстового файла.

6. После завершения работы с файлом обязательно используйте процедуру

**Close**(<имя файловой переменной>)

или

**CloseFile**(<имя файловой переменной>).

Эта процедура называется *закрыванием файла*.

### Задание 170

Запустите среду PascalABC.NET. Загрузите текст программы **file\_t1.pas**. Разберитесь в том, как она работает. Модифицируйте эту программу так, чтобы она выводила на экран содержимое любого текстового файла. Для отладки можно использовать какой-либо файл из вашей папки (в том числе

и с расширением PAS). Уберите цветное оформление текста: пусть текст выводится черным цветом.

### Задание 171

В файле **numbers.txt** хранятся числа, записанные в столбик. Составьте программу, которая позволит считать эту информацию и вычислить среднее арифметическое этих чисел. Числа могут быть любые: целые, вещественные, положительные, отрицательные. Количество чисел неизвестно!

#### *Подсказка*

Числа из текстового файла можно читать в переменные типа **Real**, **Integer**, **Word** и т. д.

### Задание 172

Загрузите текст программы **file\_t2.pas**. Используя эту программу, проверьте правильность работы программы, составленной в задании 166.

### Задание 173

Считайте с диска файл **file\_t3.pas**. Разберитесь в том, как работает программа:

- а) внимательно прочтите текст программы и комментарии;
- б) запустите программу на выполнение и введите не менее 10 чисел;
- в) считайте с диска файл **ex0.txt** и проверьте, правильно ли записались числа;
- г) модифицируйте программу так, чтобы между соседними числами был, по крайней мере, один пробел.

### Задание 174

Составьте программу – простейший текстовый редактор. Алгоритм работы достаточно прост: в диалоге вводится имя файла. Затем предлагается вводить текст. После набора строки и нажатия клавиши **<ENTER>** введенная строка записывается в файл. При введении специального

символа (это может быть «\*» или что-нибудь другое) файл закрывается и работа программы завершается.

Проверьте правильность работы вашей программы с помощью программы, разработанной при выполнении задания 166.

### **Задачи для самостоятельной работы**

1. Создайте файл, содержащий массив целых чисел. Напишите программу, вычисляющую наименьшее из этих чисел.
2. Создайте файл, содержащий массив целых чисел. Напишите программу, которая определяет количество элементов массива.
3. Создайте файл, содержащий массив вещественных чисел. Напишите программу, определяющую количество чисел, меньших среднего арифметического значения всех чисел массива.
4. Дан текстовый файл. Подсчитайте количество символов в нем.
5. Имеется текстовый файл. Удалите из него первую строку, в конце которой стоит вопросительный знак. Результат запишите в другой файл.
6. Создайте файл, содержащий массив вещественных чисел. Напишите программу, которая считывает числовые значения из файла, вычисляет значение полусуммы наибольшего и наименьшего элементов, а затем подсчитывает количество значений, не превышающих полусумму, и количество значений, больших, чем полусумма.
7. Напишите программу, которая считывает текстовый файл, выравнивает его содержимое по правой границе и выводит результат в другой текстовый файл.
8. Напишите программу, которая считывает текстовый файл, выравнивает его содержимое по центру и выводит результат в другой текстовый файл.
9. Создайте файл, содержащий фамилии студентов и их возраст. Напишите программу печати фамилий студентов, имеющих наименьший возраст.

10. Текстовый файл содержит фамилии рабочих и их среднемесячный заработок.

Напишите программу, выводящую фамилии рабочих, имеющих заработок, находящийся в указанных пользователем пределах.

☞ *Пригласите преподавателя и продемонстрируйте ему сделанные вами программы. Будьте готовы ответить на вопросы преподавателя.*

☞ *Если вы обучаетесь в условиях использования дистанционных образовательных технологий, вышлите все файлы с подготовленными вами программами преподавателю.*

## Глава 25

### РАБОТА С ДАННЫМИ ИЗ ТЕКСТОВЫХ ФАЙЛОВ

#### Задание 175

Имеется текстовый файл Shop\_1.txt, содержащий список товаров, их количество и стоимость единицы товара:

холодильники	←	название товара	
10	900	←	стоимость единицы товара
стиральный порошок			
1000	5		
микрокалькуляторы			
300	12,6		
...			

→ количество единиц товара.

Составить программу, которая создаст новый файл, содержащий в одной строке информацию о данном виде товара (название, количество единиц товара и стоимость одной единицы) плюс общую стоимость товара, а в конце файла – итоговую стоимость товаров всего этого списка.

#### Задание 176

Модифицируйте программу, разработанную в предыдущем задании, так, чтобы она решала эту же задачу, но для файла немного другой структуры (можно воспользоваться файлом shop\_2.txt):

холодильники	* 10	900
стиральный порошок	* 1000	5
микрокалькуляторы	* 300	12,6
...		

Обратите внимание на то, что название товара кончается символом \*, а затем уже идет количество и цена единицы товара.

#### Задание 177

Если у вас есть техническая возможность вывода данных на принтер, составьте универсальную программу вывода любого текстового файла на принтер.



### *Подсказка*

Для компьютера работа с принтером осуществляется так же, как и работа с текстовым файлом, имеющим стандартное имя **PRN**. Для вывода данных на принтер достаточно связать файловую переменную с этим именем:

...

```
Assign(outf,'prn');
```

...

(Надеюсь, вы догадались, что читать информацию из файла PRN не получится.)

### **Задание 178\***

Имеется файл **numbers.txt**, содержащий последовательность вещественных чисел. Составить программу, которая перепишет все числа из этого файла в другой файл в порядке возрастания (убывания).

### **Задание 179\***

Имеется два файла, содержащие списки фамилий сотрудников фондовой биржи. Фамилии в файлах расположены в алфавитном порядке. Составить программу, которая позволит на основе этих файлов создать третий, в который будут входить все фамилии, содержащиеся в первом и втором файлах. При этом фамилии должны быть также упорядочены по алфавиту.

### *Примечание*

Для отладки программы составьте два таких файла прямо в редакторе среды PascalABC.NET. Проследите, чтобы в файлах было не менее пяти строк и информация была *действительно* упорядочена по алфавиту!

### **Задание 180\***

Пусть есть текстовый файл, содержащий список (товаров, фамилий, марок автомобилей и т. д.). Составить программу, которая рассортирует этот список по алфавиту.

## **Вопросы и задания для самоконтроля**

1. Что такое файл?
2. Чем файл отличается от массива?

3. Чем характеризуется файл?
4. Как описать файловую переменную в программе на языке Паскаль?
5. Какие операции можно выполнять с файлами?
6. Какие внешние устройства могут быть использованы для хранения файлов?
7. Какие устройства могут быть использованы только для записи файлов?
8. Как связать файловую переменную, описанную в программе, с конкретным именем файла на конкретном устройстве? Напишите, как будет выглядеть эта команда:
  - а) если файл **vera.dat** на диске;
  - б) файл **raja.dat** на принтере;
  - в) файл **iren.dat** на экране терминала.
9. Какая процедура используется для подготовки файла для чтения?
10. Для чего используется процедура REWRITE?
11. Как проверить, есть ли требуемый файл на диске?
12. Какая информация может быть записана в текстовый файл?
13. Как осуществляется запись информации в текстовый файл?
14. Как в текстовый файл заносить символ «конец строки»?
15. Каково назначение функции CLOSE?
16. Как осуществляется алгоритм чтения информации из текстового файла?
17. Как дописать информацию в конец существующего текстового файла?
18. Предположим, что в файле TEMP.DAT записаны целые числа 4, 7, 2, 9, каждое из которых находится на отдельной строке. Покажите, что будет напечатано в результате выполнения показанного ниже фрагмента при следующих дополнительных условиях:
  - а) строка, содержащая число 9, является последней строкой файла и не завершается кодом «перевод строки»;

б) строка, содержащая число 9, является последней строкой файла и завершается кодом «перевод строки».

```
var datafile : text;

      x : integer;

...
assign(datafile,'temp.dat');
reset(datafile);
while not eof(datafile) do
  begin
    readln(datafile,x);
    writeln(x);
    if eof(datafile) then writeln('Конец файла')
  end;
...

```

Как исправить эту программу, чтобы она работала для случаев (а) и (б) одновременно?

### Задачи для самостоятельной работы

1. Создайте файл, содержащий фамилии студентов и их экзаменационные оценки по 5 предметам. Напишите программу печати фамилий студентов, сдавших экзамены только на «5».
2. Создайте файл, содержащий фамилии студентов. Напишите программу печати списка студентов, имеющих самые длинные фамилии.
3. Создайте файл, содержащий фамилии студентов и их экзаменационные оценки по 3 предметам. Напишите программу печати фамилий тех студентов, которые успешно сдали все экзамены.
4. В текстовом файле хранятся сведения о номерах школ учащихся, участвовавших в олимпиаде. В каждой строке сообщается количество учащихся  $N$ , каждая из  $N$  строк имеет формат:

<Фамилия> <Инициалы> <номер школы>

где <Фамилия> – строка, состоящая не более чем из 20 символов;

<Инициалы> – строка, состоящая из 4 символов (буква, точка, буква, точка);

<номер школы> – не более чем трехзначный номер.

<Фамилия> и <Инициалы>, а также <Инициалы> и <номер школы> разделены одним пробелом.


Пример входной строки:


Иванов П.С. 57

Требуется написать как можно более эффективную программу, которая будет выводить на экран информацию, из какой школы было меньше всего участников (таких школ может быть несколько). При этом необходимо вывести информацию только по школам, пославшим хотя бы одного ученика.

5. Создайте файл, содержащий фамилии студентов и их экзаменационные оценки по 3 предметам. Напишите программу печати фамилий студентов, имеющих хотя бы одну задолженность, и количество несданных экзаменов.
6. Напишите программу, которая считывает текстовый файл и определяет, содержатся ли в нем символы латинского алфавита.
7. Создайте файл, содержащий произвольные текстовые строки. Напишите программу печати строк, имеющих наименьшую длину.
8. Создайте файл, в котором содержатся семизначные номера телефонов и информация о сумме задолженности абонента данного телефона. Записи об одном абоненте могут повторяться. Напишите программу печати суммарной задолженности абонента по введенному номеру телефона.
9. Напишите программу, которая считывает исходный текст программы на языке Паскаль и подставляет номера строк (в виде комментариев) в начало каждой строки файла с исходным текстом.
10. Напишите программу, которая считывает текстовый файл и сравнивает содержащиеся в нем слова со словами, хранящимися в отдельном файле-словаре. В случае, когда слово не найдено, программа должна выдавать соответствующее сообщение.

11. Напишите программу, которая вводит с клавиатуры произвольный текст, затем записывает в первый файл закодированный текст, а во второй – дешифрованный, если известен алгоритм шифрования. Указание: код шифрования – это число, указывающее смещение букв в алфавите (например, код 3 означает, что вместо буквы «а» в зашифрованном тексте указана буква «в»).
12. Напишите программу, которая определяет количество предложений и слов в текстовом файле. Считайте, что предложением является любая последовательность символов, начинающаяся пробелом и заканчивающаяся точкой, восклицательным или вопросительным знаками.


 *Пригласите преподавателя и продемонстрируйте ему сделанные вами программы. Будьте готовы ответить на его вопросы.*

 *Если вы обучаетесь в условиях использования дистанционных образовательных технологий, вышлите все файлы с подготовленными вами программами преподавателю.*

## Глава 26

# ОСНОВЫ РАБОТЫ С ТИПИЗИРОВАННЫМИ ФАЙЛАМИ

Помимо текстовых файлов, в языке Паскаль используется другой вид файлов, который принято называть типизированным, или компонентным.

 *Типизированный* файл – это файл с заранее объявленным типом его компонент, т. е. файл с набором данных одной и той же структуры.

Для работы с нетекстовым файлом необходимо:

1) описать файловую переменную. Она описывается следующим образом:

**file of** <тип> ;

где <тип> – это тип компонент типизированного файла;

2) с помощью процедуры **Assign**(<имя\_переменной>, <имя\_файла>) связать переменную с реальным файлом на диске;

3) с помощью процедуры **Reset**(<файловая\_переменная>) открыть существующий файл для чтения и для записи;

4) с помощью процедуры **Rewrite**(<файловая\_переменная>) создать файл на диске<sup>6</sup>. При этом файл считается пустым и информация может только записываться, причем с начала файла;

5) чтение данных из файла осуществлять только с помощью процедуры **Read**, запись – **Write**.

### Задание 181

Запустите среду PascalABC.NET. Считайте файл **f\_ntx2.pas**. В файле хранится программа, записывающая в файл натуральные числа от 0 до 65 535. Выполните следующие действия:

а) разберитесь внимательно в тексте программы. Запустите программу на выполнение. Проверьте правильность работы программы при любых

---

<sup>6</sup> Если использовать **Rewrite** для существующего файла, его содержимое будет уничтожено и с ним можно будет работать точно так же, как с пустым.

данных (числовых и нечисловых). При обнаружении ошибки исправьте ее;

- б) запустите программу на выполнение и введите 10 раз пару чисел: 11 и 13. Это понадобится для выполнения следующего задания.

### Задание 182

Считайте файл **f\_ntx1.pas**. В файле хранится программа, читающая информацию, создаваемую программой из предыдущего задания. Выполните следующее:

- а) программа испорчена – вместо одной из строк программы поставлен комментарий из звездочек. Восстановите эту строку (ее смысл поясняется комментарием, расположенным правее). Проверьте правильность работы программы (данные, хранящиеся в файле **stud.dat**, известны);
- б) измените программу так, чтобы она после вывода всех чисел вывела среднее арифметическое этих чисел;
- в) выполните программу **f\_ntx2.pas** и введите произвольный набор чисел;
- г) измените программу **f\_ntx1.pas** так, чтобы на экран выводились числа, значения которых не меньше заданного числа.

### Задание 183

Составьте программу, которая создает сразу два файла. Первый файл (назовем его **fam.dat**) должен содержать название товара, второй (**costs.dat**) – его стоимость. Ввод данных с экрана должен осуществляться последовательно: название товара, его стоимость, название товара, его стоимость...

### Задание 184

Пусть дан файл **fam.dat**, содержащий название товара, и **costs.dat**, содержащий цены этих товаров. Составьте программу, которая определяет:

- а) название самого дорогого товара;
- б) название самого дешевого товара;
- в) среднюю стоимость товаров.

## Задание 185

Имеются две программы, записывающие одну и ту же числовую информацию в файлы:

- |                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| а) <b>program a1;</b><br><b>var x:real:= 3569.45603;</b><br><b>f:text;</b><br><b>begin</b><br><b>assign(f,'num1.dat');</b><br><b>rewrite(f);</b><br><b>repeat</b><br><b>writeln(f,x);</b><br><b>x+=1</b><br><b>until x&gt;3579.45603;</b><br><b>close(f)</b><br><b>end.</b> | б) <b>program a2;</b><br><b>var x:real:= 3569.45603;</b><br><b>f:file of real;</b><br><b>begin</b><br><b>assign(f,'num1.dat');</b><br><b>rewrite(f);</b><br><b>repeat</b><br><b>writeln(f,x);</b><br><b>x+=1</b><br><b>until x&gt;3579.45603;</b><br><b>close(f)</b><br><b>end.</b> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Определите размеры файлов в байтах. Какой из файлов будет занимать больше места на диске?

## Задачи для самостоятельной работы

1. Создайте программу, которая с помощью датчика случайных чисел создаст файл вещественных чисел. Программа должна заменить в файле все элементы на их квадраты.
2. Дан файл, содержащий различные даты. Каждая дата – это число, месяц, год. Найти год с наименьшим номером.
3. Дан файл, содержащий сведения об учениках школы. Каждая запись включает имя, фамилию, название класса (год обучения и букву), в котором ученик учится. Составьте программу, с помощью которой можно:  
выяснить, имеются ли в школе однофамильцы;  
определить, имеются ли однофамильцы в параллельных классах;  
то же, в каком-нибудь классе;  
выяснить, в каких классах насчитывается менее 25 учеников;  
выяснить, на сколько человек в девятых классах больше, чем в одиннадцатых;  
сформировать в текстовом файле список учеников заданного класса.
4. Дан файл, содержащий сведения об автомобилях. Каждая запись включает марку автомобиля, его номера и фамилии владельца. Составьте программу, с помощью которой можно определить:



фамилии владельцев и номера автомобилей данной марки;  
количество автомобилей каждой марки.

5. Составьте программу, которая создает типизированный файл, содержащий информацию о работниках института. Помимо ввода информации программа должна обеспечить вывод:

среднего стажа работы в институте;

фамилий работников, работающих на должности инженера;


работников, фамилии которых начинаются с заданной буквы;


фамилий работников, у которых оклад больше (меньше) заданного пользователем;

фамилий работников пенсионного и предпенсионного возраста.

6. Имеется типизированный файл, содержащий данные о работающих в фирме: фамилию, имя, отчество, адрес (улица, дом, квартира) и дату поступления на работу (месяц, год). Во второй файл запишите данные только о тех из них, кто на сегодняшний день проработал не менее 5 лет. Затем расположите в файле записи в алфавитном порядке по фамилиям.

7. Имеется типизированный файл, содержащий данные об учениках школы: фамилию, имя, отчество, адрес (улица, дом, квартира), класс. Во второй файл запишите все данные об учениках определенного класса. Затем расположите в файле записи в алфавитном порядке по фамилиям. Выведите содержимое файла, выделяя тех, кто живет на заданной улице.

 *Пригласите преподавателя и продемонстрируйте ему сделанные вами программы. Будьте готовы ответить на вопросы преподавателя.*

 *Если вы обучаетесь в условиях использования дистанционных образовательных технологий, вышлите все файлы с подготовленными вами программами преподавателю.*

